

Minería de datos y Patrones

Version 2025-I

Aprendizaje supervisado – Intro

Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingeniería Electronica
Universidad Popular del Cesar

Aprendizaje supervisado – Intro

Esquema: Aprendizaje supervisado

Marco global

Generalidades

Contexto teórico

Marco matemático

Ejemplos

Aprendizaje

Función de costo

Errores, complejidad y
sobre-aprendizaje

Regularización

Optimización

Evaluación de la predicción

Métricas de clasificación

Métricas de regresión

Muestro para validación

Marco global

Outline : Marco global

Marco global

Generalidades

Contexto teorico

Aprendizaje

Evaluacion de la predicci' on

Outline : Generalidades

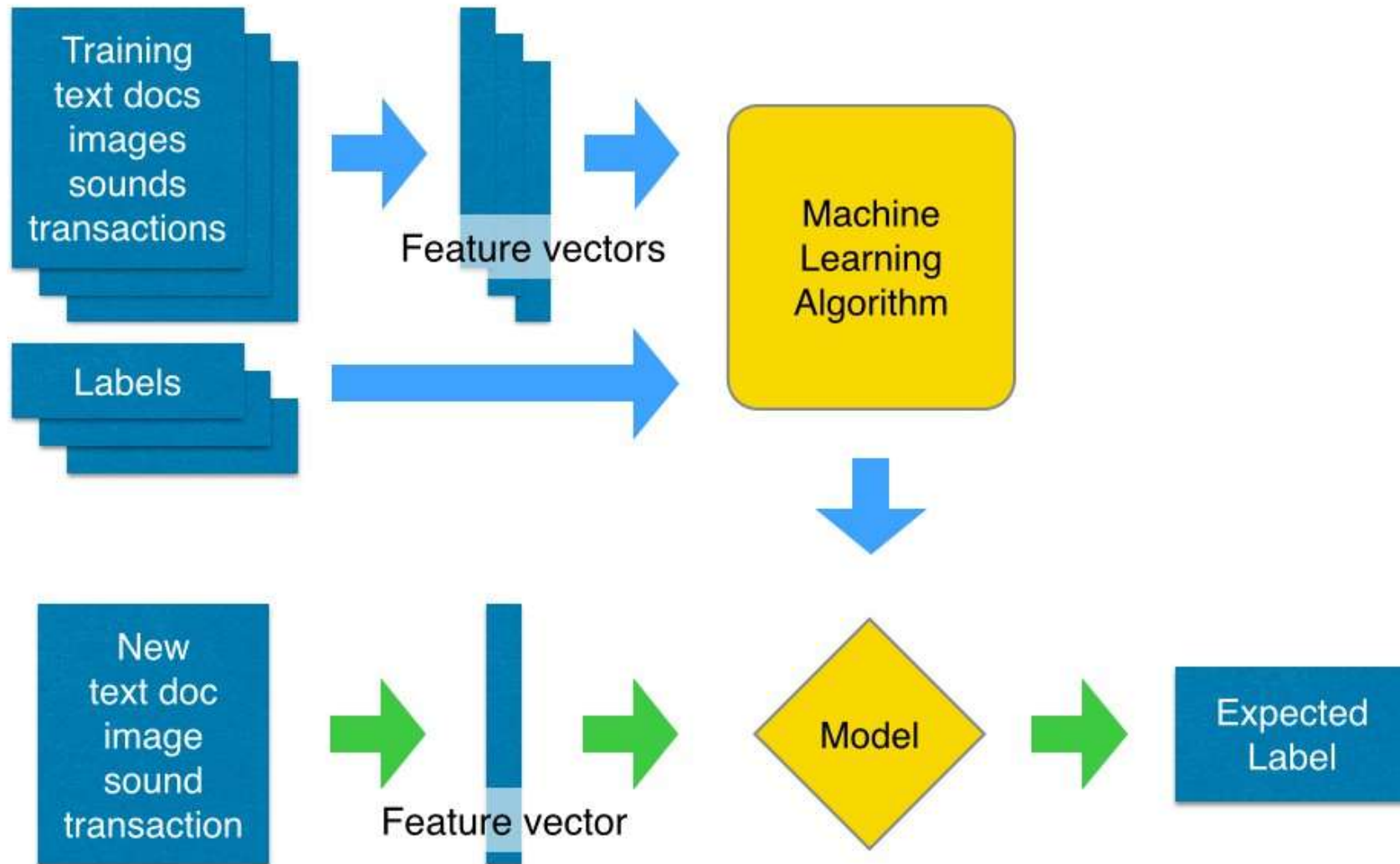
Marco global

Generalidades

Contexto
teorico
Aprendizaje

Evaluacion de la predicci' on

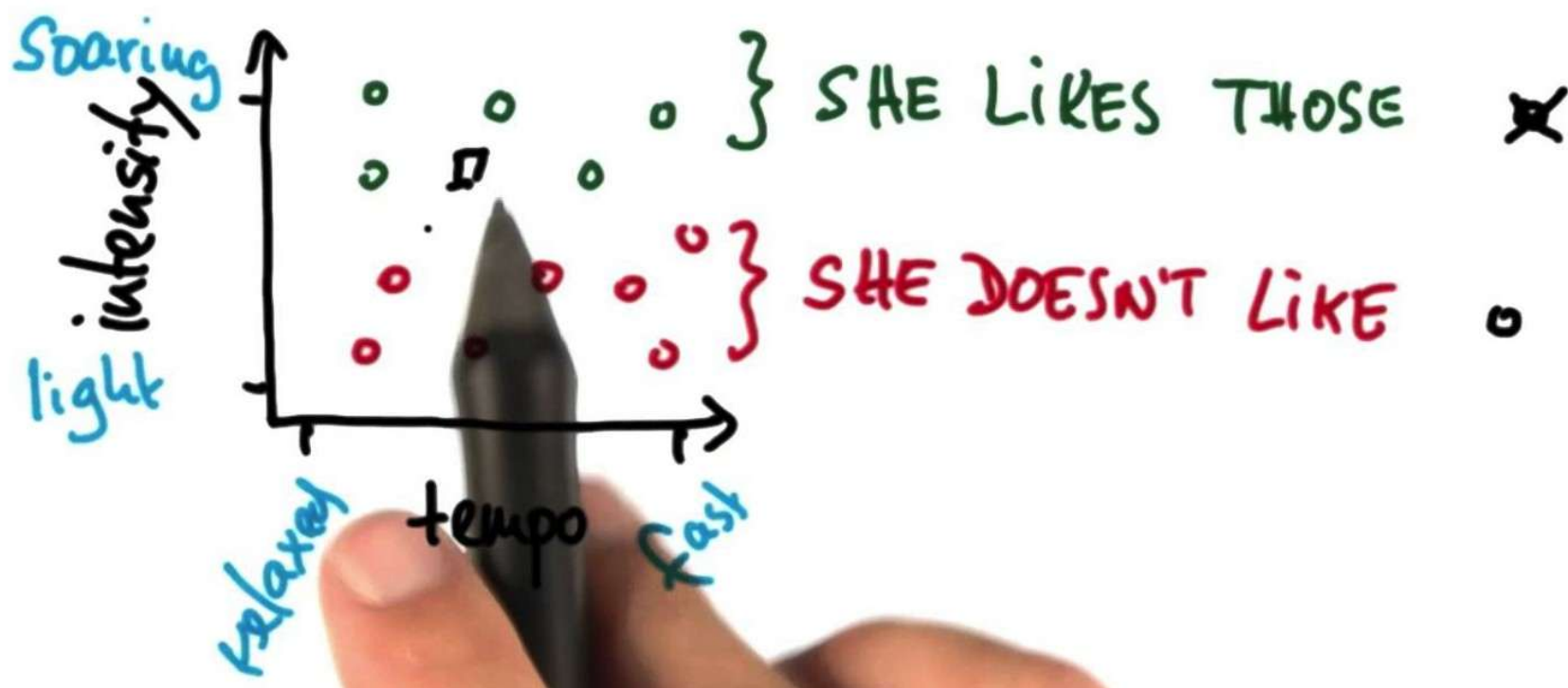
Aprendizaje supervisado



Predictive Modeling Data Flow

Descriptores y etiquetas

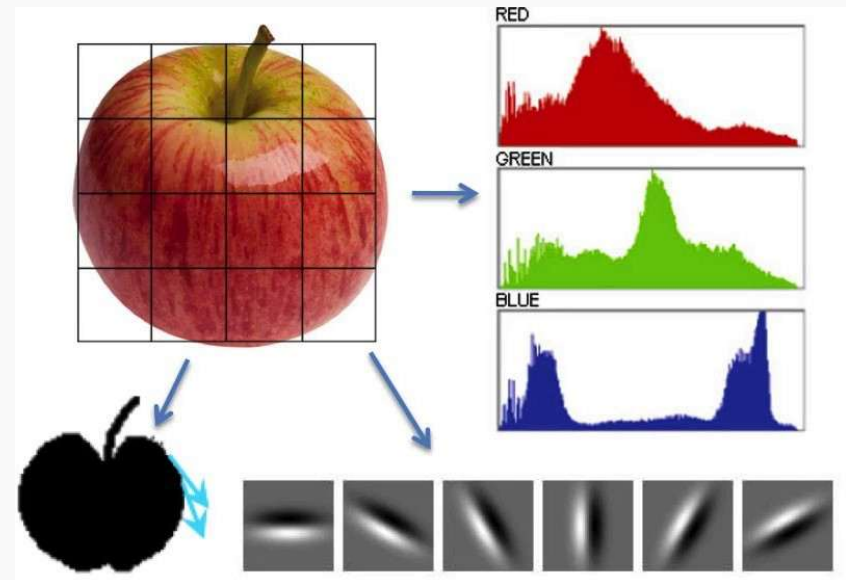
FEATURES AND LABELS



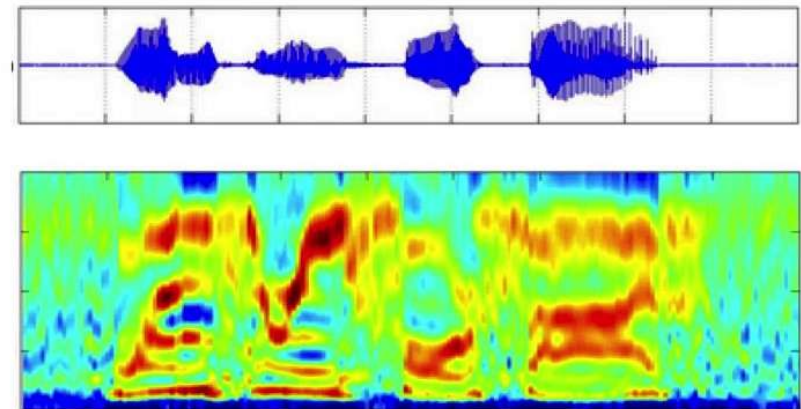
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Figure 1: Un base de datos estructurados

- Imágenes: Histograma de colores



- Sonidos: Representación tiempo-frecuencia



I Tener datos etiquetados

II Extraer los descriptores = transformar documentos en vectores

III Crear un modelo matemático f_{θ}

VI Implementar una función de costo (error) ℓ a minimizar

V Encontrar los parámetros ϑ^* de manera que $\ell(f_{\theta^*}(\mathbf{X}_i), Y_i)$ sea pequeño

VI Probar f_{θ^*} en nuevos datos con una métrica de evaluación adecuada

I Tener datos etiquetados

- Conjunto de datos de tamaño n , $D_n = \{(\text{Doc}_i, Y_i), i = 1..n\}$
- Doc es una muestra (por ejemplo: una persona)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

II **Extraer los descriptores** = transformar documentos en vectores

III **Crear un modelo matemático** f_θ

VI **Implementar una función de costo (error) ℓ a minimizar**

V **Encontrar los parámetros ϑ^* de manera que $\ell(f_{\theta^*}(\mathbf{X}_i), Y_i)$ sea pequeño**

VI **Probar f_{θ^*} en nuevos datos con una métrica de evaluación adecuada**

I Tener datos etiquetados

- Conjunto de datos de tamaño n , $D_n = \{(\text{Doc}_i, Y_i), i = 1..n\}$
- Doc es una muestra (por ejemplo: una persona)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

II Extraer los descriptores = transformar documentos en vectores

- \mathbf{X} es un vector de observaciones (por ejemplo: edad, sexo, salario)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

III Crear un modelo matemático f_θ

VI Implementar una función de costo (error) ℓ a minimizar

V Encontrar los parámetros ϑ^* de manera que $\ell(f_{\theta^*}(\mathbf{X}_i), Y_i)$ sea pequeño

VI Probar f_{θ^*} en nuevos datos con una métrica de evaluación adecuada

I. Tener datos etiquetados

- Conjunto de datos de tamaño n , $D_n = \{(\text{Doc}_i, Y_i), i = 1..n\}$
- Doc es una muestra (por ejemplo: una persona)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

II. Extraer los descriptores = transformar documentos en vectores

- \mathbf{X} es un vector de observaciones (por ejemplo: edad, sexo, salario)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

III. Crear un modelo matemático f_θ

- Modelo f_θ tal que $f_\theta(\mathbf{X})$ está cerca de Y (para regresión)
- θ es el conjunto de parámetros del modelo matemático

VI Implementar una función de costo (error) ℓ a minimizar

V Encontrar los parámetros ϑ^* de manera que $\ell(f_{\vartheta^*}(\mathbf{X}_i), Y_i)$ sea pequeño

VI Probar f_{ϑ^*} en nuevos datos con una métrica de evaluación adecuada

I. Tener datos etiquetados

- Conjunto de datos de tamaño n , $D_n = \{(\text{Doc}_i, Y_i), i = 1..n\}$
- Doc es una muestra (por ejemplo: una persona)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

II. Extraer los descriptores = transformar documentos en vectores

- \mathbf{X} es un vector de observaciones (por ejemplo: edad, sexo, salario)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

III. Crear un modelo matemático f_θ

- Modelo f_θ tal que $f_\theta(\mathbf{X})$ esté cerca de Y (para regresión)
- θ es el conjunto de parámetros del modelo matemático

VI Implementar una función de costo (error) ℓ a minimizar

- Cuanto más se equivoque el modelo, mayor será el costo
- En general, se desea tener un costo pequeño

V Encontrar los parámetros ϑ^* de manera que $\ell(f_{\vartheta^*}(\mathbf{X}_i), Y_i)$ sea pequeño

VI Probar f_{ϑ^*} en nuevos datos con una métrica de evaluación adecuada

I. Tener datos etiquetados

- Conjunto de datos de tamaño n , $D_n = \{(Doc_i, Y_i), i = 1..n\}$
- Doc es una muestra (por ejemplo: una persona)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

II. Extraer los descriptores = transformar documentos en vectores

- \mathbf{X} es un vector de observaciones (por ejemplo: edad, sexo, salario)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

III. Crear un modelo matemático f_θ

- Modelo f_θ tal que $f_\theta(\mathbf{X})$ esté cerca de Y (para regresión)
- θ es el conjunto de parámetros del modelo matemático

VI Implementar una función de costo (error) ℓ a minimizar

- Cuanto más se equivoque el modelo, mayor será el costo
- En general, se desea tener un costo pequeño

V Encontrar los parámetros ϑ^* de manera que $\ell(f_{\vartheta^*}(\mathbf{X}_i), Y_i)$ sea pequeño

- $\theta^* = \arg \min_{\theta} \sum_i \ell(f_\theta(\mathbf{X}_i), Y_i)$

VI Probar f_{θ^*} en nuevos datos con una métrica de evaluación adecuada

I. Tener datos etiquetados

- Conjunto de datos de tamaño n , $D_n = \{(\text{Doc}_i, Y_i), i = 1..n\}$
- Doc es una muestra (por ejemplo: una persona)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

II. Extraer los descriptores = transformar documentos en vectores

- \mathbf{X} es un vector de observaciones (por ejemplo: edad, sexo, salario)
- Y son las etiquetas (por ejemplo: monto del préstamo concedido)

III. Crear un modelo matemático f_θ

- Modelo f_θ tal que $f_\theta(\mathbf{X})$ esté cerca de Y (para regresión)
- θ es el conjunto de parámetros del modelo matemático

VI Implementar una función de costo (error) ℓ a minimizar

- Cuanto más se equivoque el modelo, mayor será el costo
- En general, se desea tener un costo pequeño

V Encontrar los parámetros θ^* de manera que $\ell(f_{\theta^*}(\mathbf{X}_i), Y_i)$ sea pequeño

$\theta^* = \arg \min_{\theta} \sum_i \ell(f_{\theta}(\mathbf{X}_i), Y_i)$

VI Probar f_{θ^*} en nuevos datos con una métrica de evaluación adecuada

Outline : Contexto teórico

Marco global

Generalidades

Contexto
teórico

Marco matemático

Ejemplos

Aprendizaje

Evaluación de la predicción

Marco matemático

- Medida de entrada: $\mathbf{X} = (X^{(1)}, X^{(2)}, \dots, X^{(d)}) \in X$
- Medida de salida: $Y \in Y$
- $(X, Y) \sim \mathbf{P}$ donde \mathbf{P} es desconocido
- Conjunto de entrenamiento: $D_n = (\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$
- A menudo:
 - $\mathbf{X} \in \mathbb{R}^d$ y $Y \in \{1, \dots, C\}$ para una tarea de clasificación
 - $\mathbf{X} \in \mathbb{R}^d$ y $Y \in \mathbb{R}$ para una tarea de regresión

Un clasificador es una función en $F = \{f : X \rightarrow Y\}$

(medible)

Objetivo

Construir un clasificador \hat{f} **satisfactorio** utilizando los datos de entrenamiento. **Atención:** \hat{f} **depende de** D_n .

Aprendizaje, tarea, medida de rendimiento

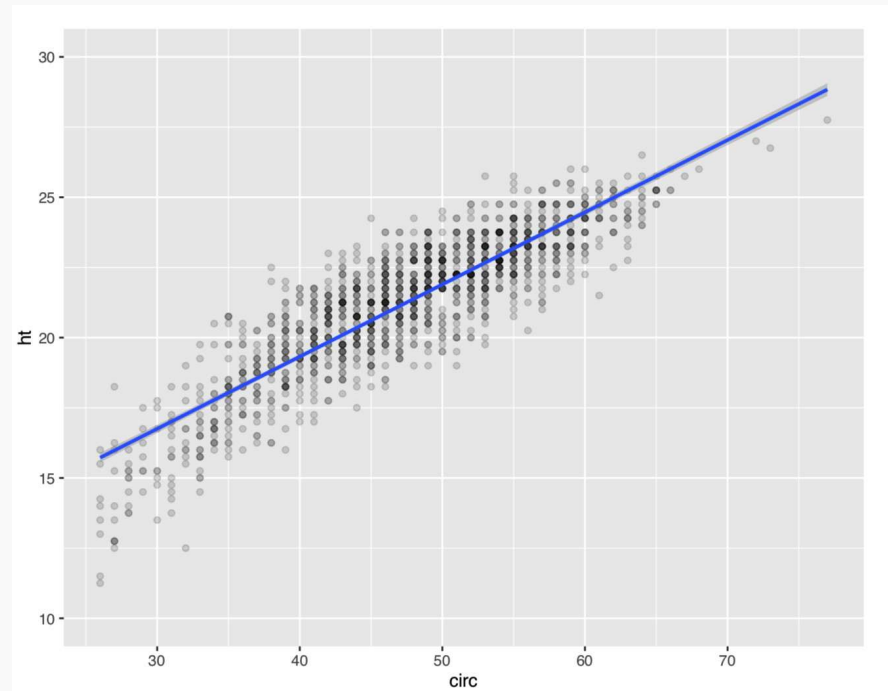
- **Conjunto de entrenamiento:** $D_n = (\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)$
- **Clasificador:** $f : X \rightarrow Y$ (medible).
- **Función de costo/pérdida:** $\ell(Y, f(\mathbf{X}))$ mide la calidad de la *predicción* de f en relación con Y
- **Riesgo:**

$$R(f) = E[\ell(Y, f(\mathbf{X}))]$$

Objetivo

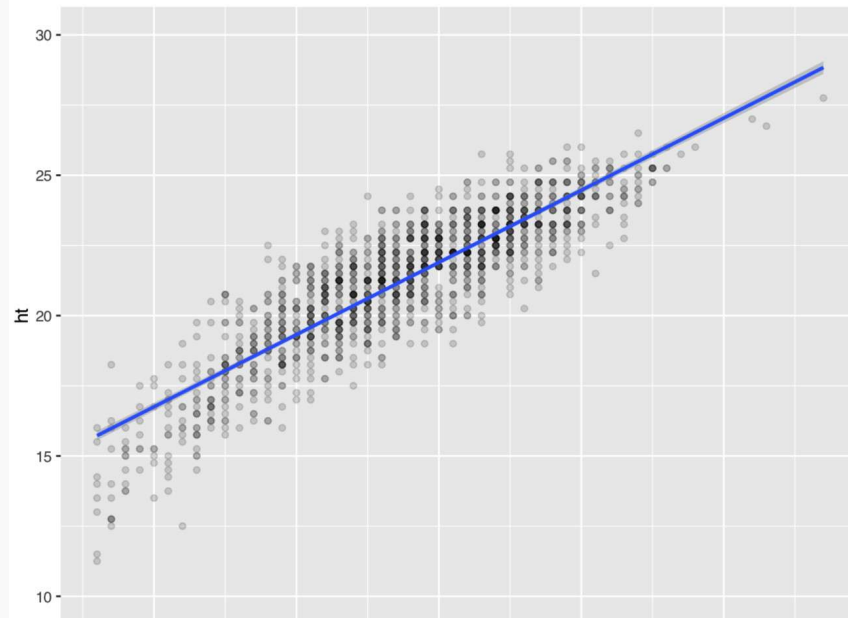
Aprender a construir un **clasificador** $f \in F$ desde los datos de entrenamiento D_n de manera que el **riesgo** $R(f)$ sea **pequeño o en promedio** en relación con D_n .

Ejemplo: Eucalipto



- BDD simple y clásica
- Predecir el tamaño en función de la circunferencia
- **X = circ**
- $Y = ht$

Ejemplo: Eucalipto



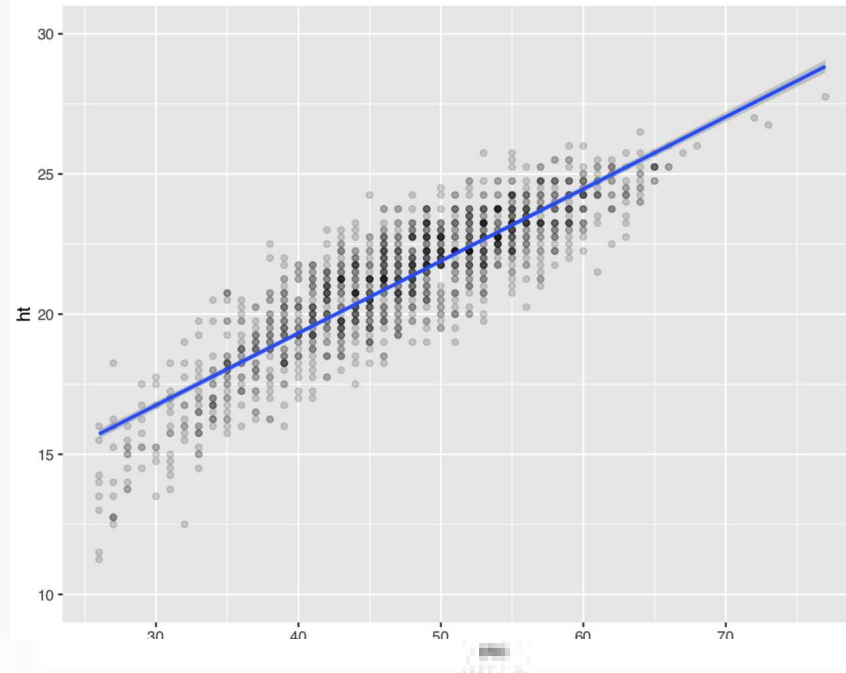
Modelo Lineal

- Modelo paramétrico:

$$f_{\beta}(\text{circ}) = \beta_1 + \beta_2 \text{circ} = \hat{ht}$$

- ¿Cómo elegir $\beta = (\beta_1, \beta_2)$?

Ejemplo: Mínimos cuadrados



Metodología

- Naturalmente:

$$\sum_{i=1}^n |Y_i - f_{\beta}(\mathbf{X}_i)|^2 = \sum_{i=1}^n |ht_i - f_{\beta}(\text{circ}_i)|^2 = \sum_{i=1}^n |ht_i - (\beta_1 + \beta_2 \text{circ}_i)|^2$$

- Elección de β que minimiza este criterio:

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^2} \sum_{i=1}^n |ht_i - (\beta_1 + \beta_2 \text{circ}_i)|^2$$

Regresión Lineal

- **Modelo estadístico:** (circ_i, ht_i) i.i.d. con la misma ley que un (circ, ht) genérico
- **Criterio de rendimiento:** Buscar un f con un error promedio bajo

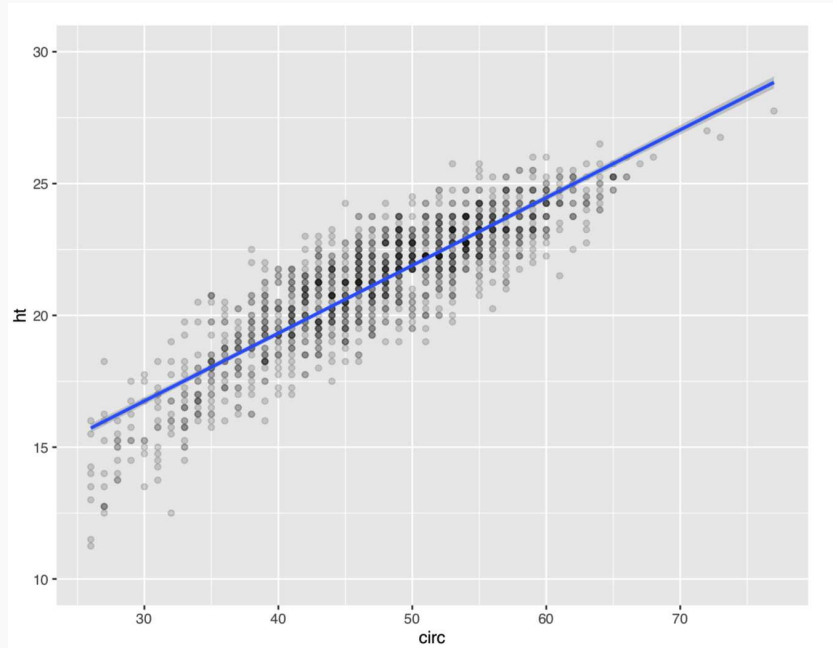
$$E[|ht - f(\text{circ})|^2]$$

- **Criterio empírico:** Reemplazar la ley desconocida por su contraparte empírica

$$\frac{1}{n} \sum_{i=1}^n |ht_i - f(\text{circ}_i)|^2$$

- **Elegir un clasificador:** no tomar un modelo demasiado complejo, restringirse al modelo más simple que dé buenos resultados (Ej: gran red neuronal/demasiadas características para muy pocos ejemplos)
- **Aprender el modelo:** Optimizar sobre los datos

Ejemplo: ¿Qué grado elegir ?



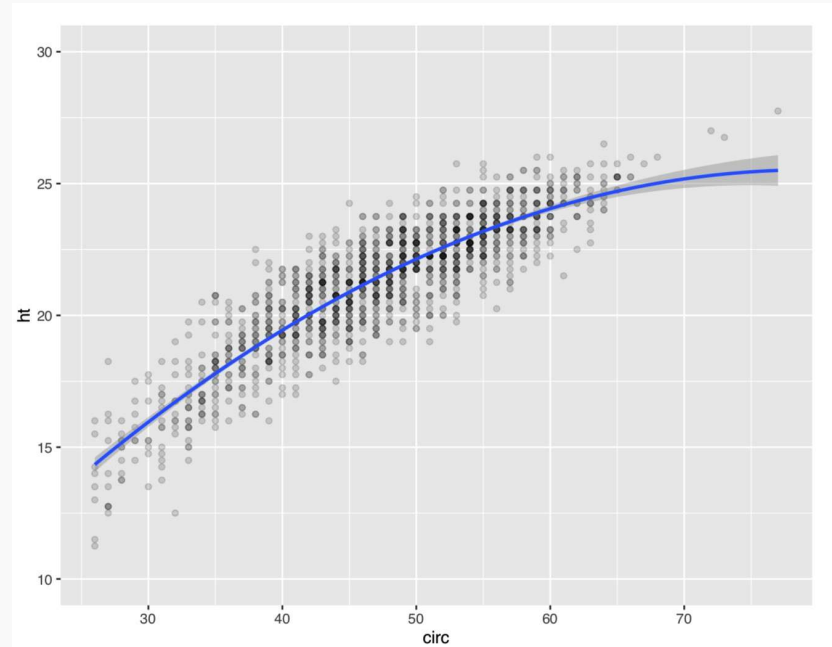
Modelos

- Un grado polinomial d que aumenta equivale a una **complejidad de modelo que aumenta** :

$$\beta \in \mathbb{R}^d = \text{plus de parametres}$$

- Este modelo ya no se adapta a los datos que se le muestran.
- **Sobreajuste** : se adapta demasiado a D_n y ya no generaliza

Ejemplo: ¿Qué grado elegir ?



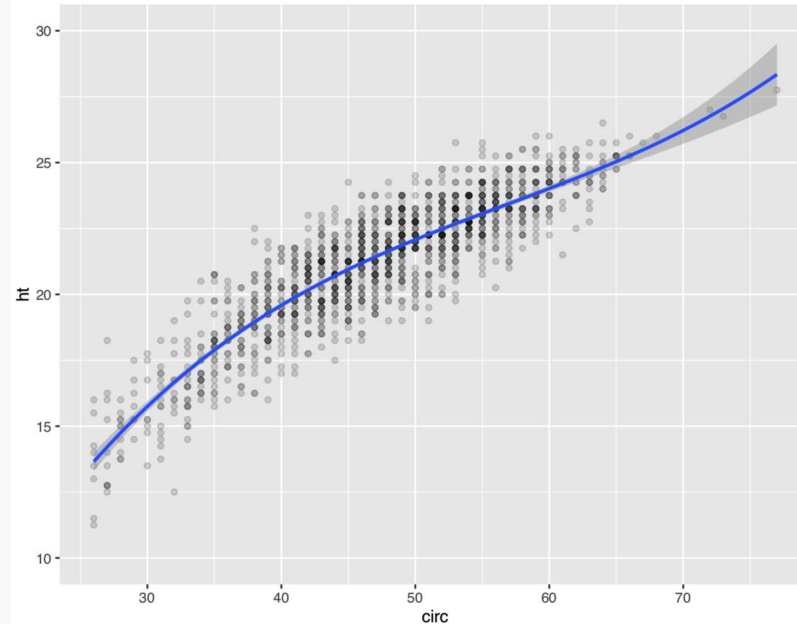
Modelos

- Un grado polinomial d que aumenta equivale a una **complejidad de modelo que aumenta** :

$$\theta \in \mathbb{R}^d = \text{plus de paramètres}$$

- Este modelo ya no se adapta a los datos que se le muestran.
- **Sobreajuste** : se adapta demasiado a D_n y ya no generaliza

Ejemplo: ¿Qué grado elegir ?



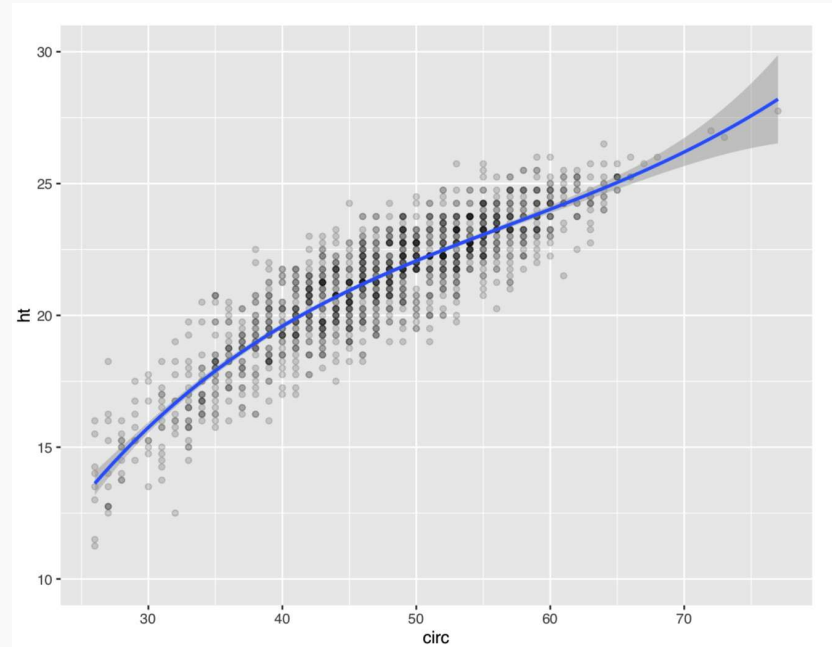
Modelos

- Un grado polinomial d que aumenta equivale a una **complejidad de modelo que aumenta** :

$$\theta \in \mathbb{R}^d = \text{plus de param`etres}$$

- Este modelo ya no se adapta a los datos que se le muestran.
- **Sobreajuste** : se adapta demasiado a D_n y ya no generaliza

Ejemplo: ¿Qué grado elegir ?



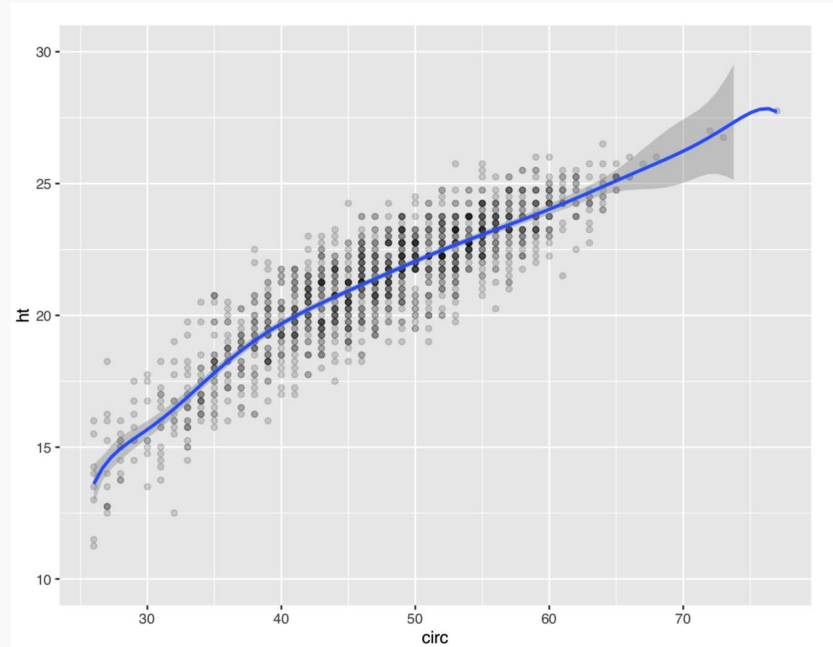
Modelos

- Un grado polinomial d que aumenta equivale a una **complejidad de modelo que aumenta** :

$$\theta \in \mathbb{R}^d = \text{plus de param`etres}$$

- Este modelo ya no se adapta a los datos que se le muestran.
- **Sobreajuste** : se adapta demasiado a D_n y ya no generaliza

Ejemplo: ¿Qué grado elegir ?



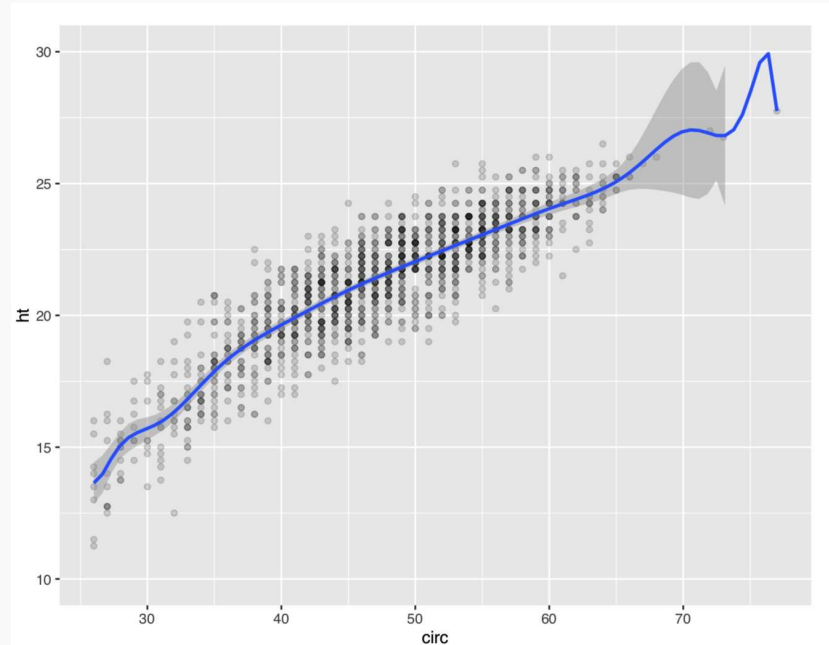
Modelos

- Un grado polinomial d que aumenta equivale a una **complejidad de modelo que aumenta** :

$$\theta \in \mathbb{R}^d = \text{plus de paramètres}$$

- Este modelo ya no se adapta a los datos que se le muestran.
- **Sobreajuste** : se adapta demasiado a D_n y ya no generaliza

Ejemplo: ¿Qué grado elegir ?



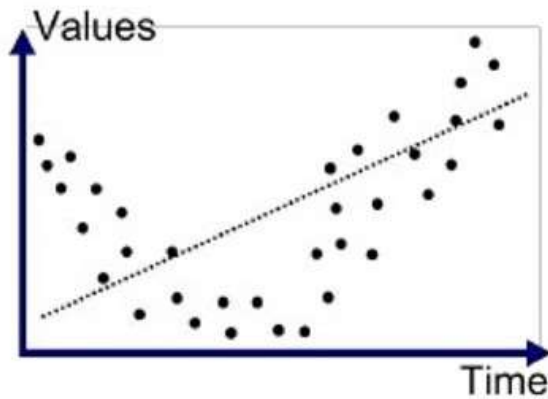
Modelos

- Un grado polinomial d que aumenta equivale a una **complejidad de modelo que aumenta** :

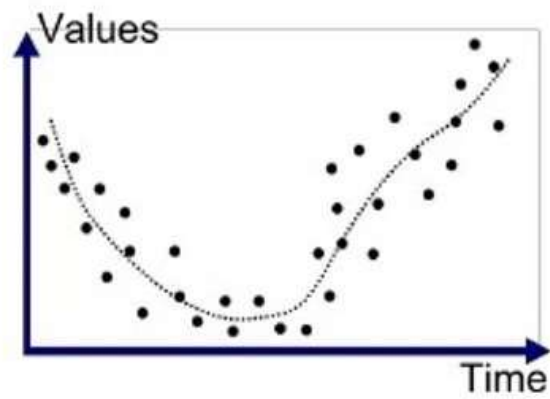
$$\theta \in \mathbb{R}^d = \text{plus de paramètres}$$

- Este modelo ya no se adapta a los datos que se le muestran.
- **Sobreajuste** : se adapta demasiado a D_n y ya no generaliza

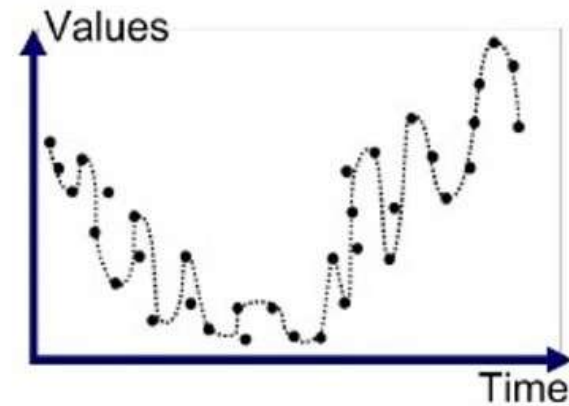
Ejemplo: ¿Qué grado elegir ?



Underfitted



Good Fit/Robust



Overfitted

Complejidad

- Si el modelo es demasiado simple, entonces ya no sigue los datos
- Si el modelo es demasiado complejo, el modelo aprende todas las irregularidades del dataset D_n
- Ejemplo: si el modelo es el de la curva del medio más una componente de ruido no tenida en cuenta en las variables, el modelo de la derecha aprende ese ruido

Outline : Aprendizaje

Marco global

Generalidades

Contexto teórico

Aprendizaje

Función de costo

Errores, complejidad y
sobre-aprendizaje

Regularización

Optimización

Evaluación de la
predicción

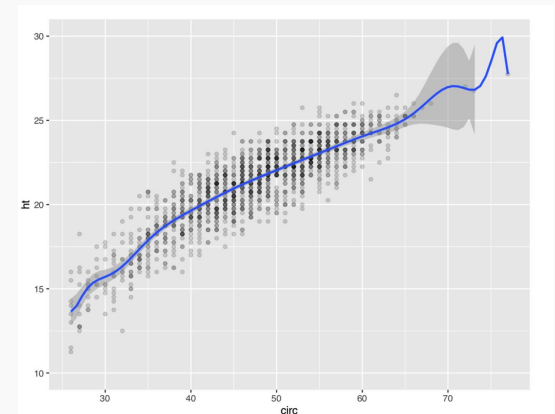
Aprendizaje supervisado

Diferentes conceptos

- **Datos etiquetados:** Regresión o Clasificación
- **Extracción de características:** Tono, Intensidad, Tempo o Edad, Salario, Género, etc.
- **Modelo f_θ :** SVM, Regresión Logística, Bosque Aleatorio, CNN
- **Función de costo** a optimizar: Pérdida de Bisagra, Pérdida de Entropía Cruzada, Pérdida Logística, Pérdida Cuadrada, etc.
- **Algoritmo de optimización:** Adam, SGD, BFGS, etc.
- **Métrica de evaluación:** Recall, Precisión, Mínimos Cuadrados, etc.

En el caso anterior:

- Descriptores y etiquetas:
- Modelo:
- Funcion de costo:
- Optim:
- Métrica de evaluación:



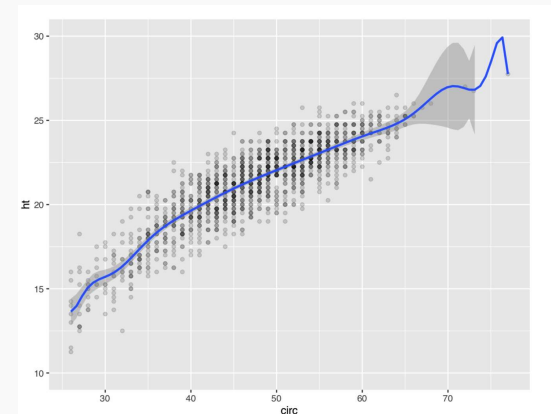
Aprendizaje supervisado

Diferentes conceptos

- **Datos etiquetados:** Regresión o Clasificación
- **Extracción de características:** Tono, Intensidad, Tempo o Edad, Salario, Género, etc.
- **Modelo f_θ :** SVM, Regresión Logística, Bosque Aleatorio, CNN
- **Función de costo** a optimizar: Pérdida de Bisagra, Pérdida de Entropía Cruzada, Pérdida Logística, Pérdida Cuadrada, etc.
- **Algoritmo de optimización:** Adam, SGD, BFGS, etc.
- **Métrica de evaluación:** Recall, Precisión, Mínimos Cuadrados, etc.

En el caso anterior:

- Descriptores y etiquetas: Circunferencia y longitud
- Modelo: Polinomio de grado 6
- Funcion de costo: Pérdida Cuadrada
- Optim: Descenso de Gradiente Estocástico (SGD)
- Métrica de evaluación: Mínimos Cuadrados



Entrenamiento: función de costo ℓ

Principio

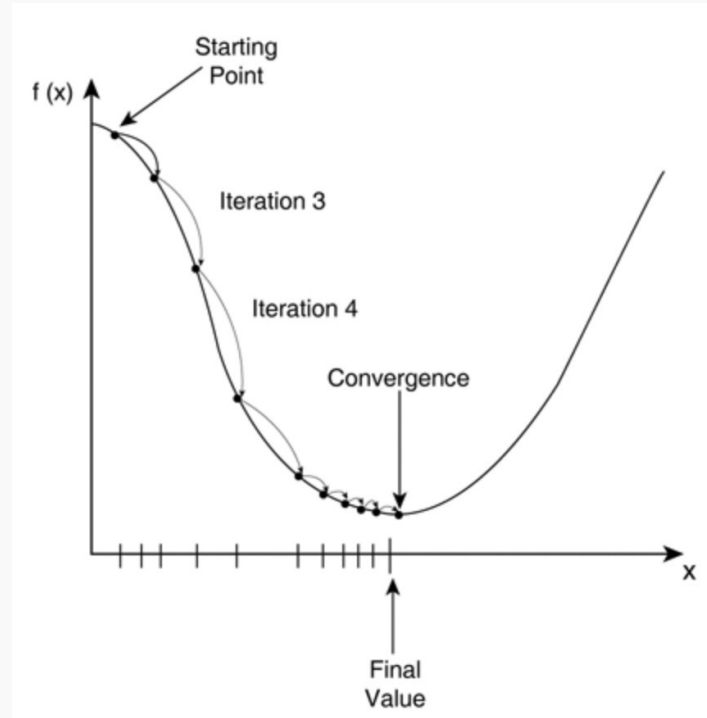
- Función que **penaliza** cuando el modelo comete errores
- Minimizar esta función sobre el conjunto de entrenamiento (riesgo empírico) para encontrar parámetros del modelo satisfactorios:

$$f_{\hat{\theta}} = \arg \min_{f_{\theta}, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f_{\theta}(\mathbf{X}_i))$$

- Convexidad: generalmente se reemplaza la función de costo $\ell^{0/1}$ por una función convexa ℓ' más fácil de optimizar

- Expresa el error desde una perspectiva **numérica**
- Transmite al algoritmo de aprendizaje lo que es importante y tiene sentido para la tarea
- Debe ser una función que se pueda optimizar eficientemente (convexa). **La función $\ell^{0/1} = 1f(\mathbf{X}) = Y$ no es utilizable** (ni siquiera continua).

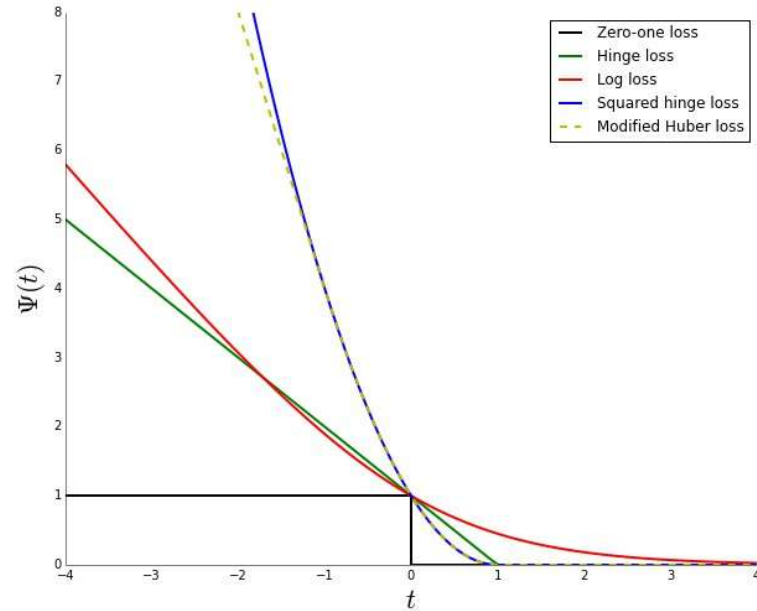
Función de costo y convexidad



Ejemplos de funciones de costo clásicas

- Logística (*Softmax*): $\ell'(Y, f(\mathbf{X})) = \log(1 + \exp^{-Yf(\mathbf{X})})$
- Bisagra: $\ell'(Y, f(\mathbf{X})) = (1 - Yf(\mathbf{X}))_+$
- Exponencial: $\ell'(Y, f(\mathbf{X})) = \exp^{-Yf(\mathbf{X})}$
- Cross-Entropy: $\ell'(Y, f(\mathbf{X})) = -(Y \ln(f(\mathbf{X})) + (1 - Y) \ln(1 - f(\mathbf{X})))$

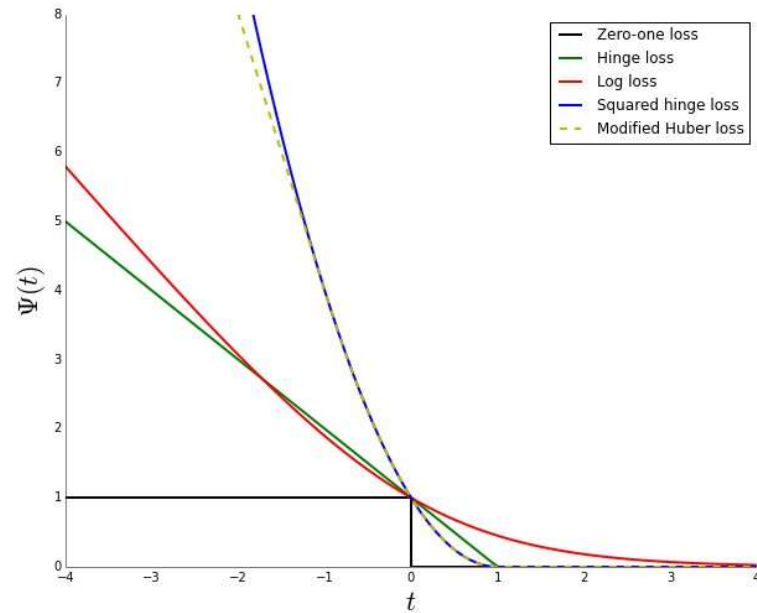
Función de costo y convexidad



Ejemplo de funciones de costo clásicas

- Logística (*Softmax*): $\ell'(Y, f(\mathbf{X})) = \log(1 + \exp^{-Yf(\mathbf{X})})$
- Exponencial: $\ell'(Y, f(\mathbf{X})) = \exp^{-Yf(\mathbf{X})}$
- con $Y = \pm 1$, queremos que $f(\mathbf{X})$ sea muy **pos/neg** para $Y = +1/-1$
- Si $f(\mathbf{X}) = \text{signo}(Y)$, entonces $\exp^{-Yf(\mathbf{X})}$ es pequeño

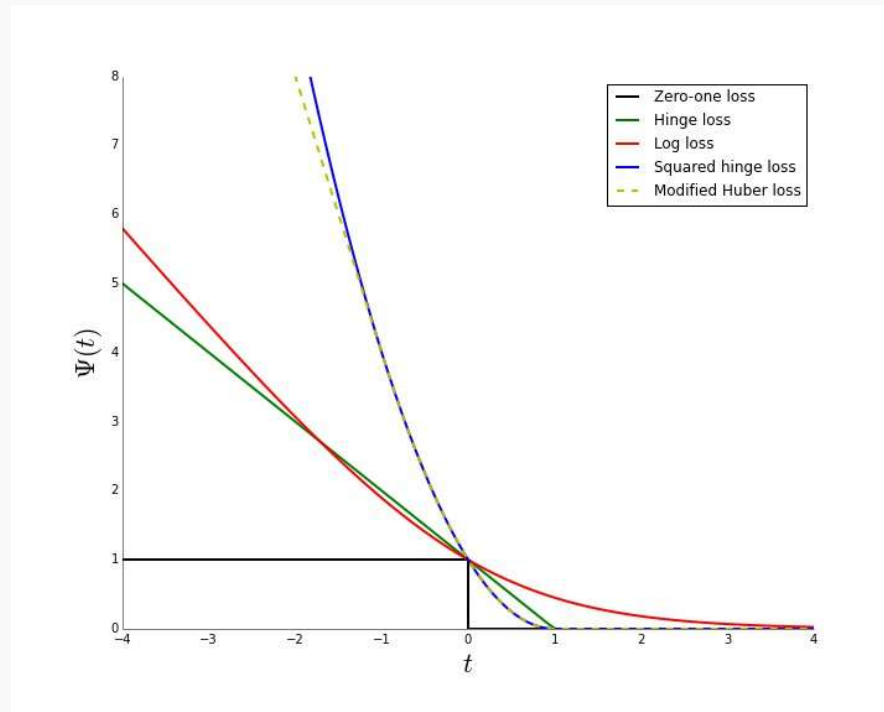
Función de costo y convexidad



Ejemplo de funciones de costo clásicas

- Cuadrada: $\ell'(Y, f(\mathbf{X})) = (1 - Yf(\mathbf{X}))^2$
- con $Y = \pm 1$, queremos exactamente $Y = f(\mathbf{X})$ (signo y amplitud)

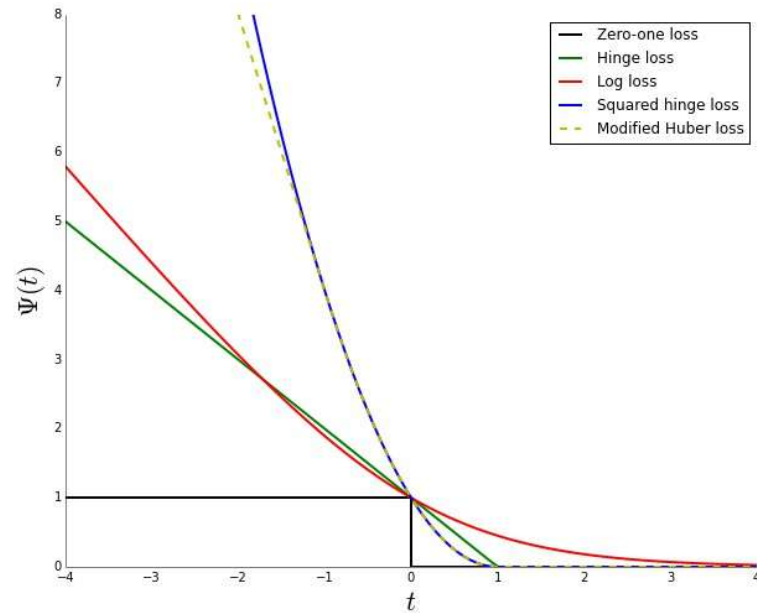
Función de costo y convexidad



Ejemplo de funciones de costo clásicas

- Bisagra: $\ell'(Y, f(\mathbf{X})) = (1 - Yf(\mathbf{X}))_+$
- con $Y = \pm 1$
- Si $f(\mathbf{X}) = \text{signo}(Y)$, y $|f(\mathbf{X})| > 1$ entonces $Yf(\mathbf{X}) > 1$

Función de costo y convexidad



Ejemplo de funciones de costo clásicas

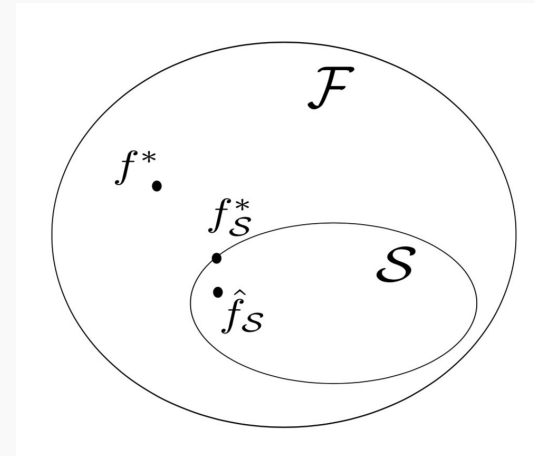
- Cross-Entropy: $\ell'(Y, f(\mathbf{X})) = -(Y \ln(f(\mathbf{X})) + (1 - Y) \ln(1 - f(\mathbf{X})))$
- con $Y \in \{0, 1\}$, queremos $f(\mathbf{X}) = Y$

-

$$\ell'(Y, f(\mathbf{X})) = \begin{cases} -\ln(1 - f(\mathbf{X})) & \text{si } Y = 0 \\ -\ln(f(\mathbf{X})) & \text{si } Y = 1 \end{cases}$$

Complejidad y modelos

- $F = \{f : \text{funciones medibles } X \rightarrow Y\}$
- Mejor solución $f^* = \arg \min_{f \in F} R(f)$
- Clase de funciones $S \subset F$ utilizadas como modelos
- Objetivo ideal en S : $f_S^* = \arg \min_{f \in S} R(f)$
- Estimación obtenida en S : se obtiene \hat{f}_S tras un entrenamiento

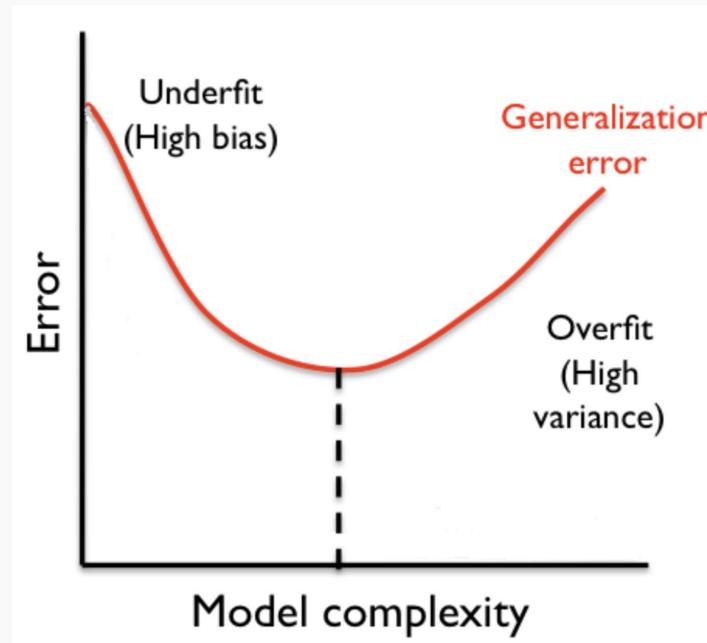


Error de aproximación y error general

$$\mathcal{R}(\hat{f}_S) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f_S^*) - \mathcal{R}(f^*)}_{\text{error de aproximación}} + \underbrace{\mathcal{R}(\hat{f}_S) - \mathcal{R}(f_S^*)}_{\text{error de estimación}}$$

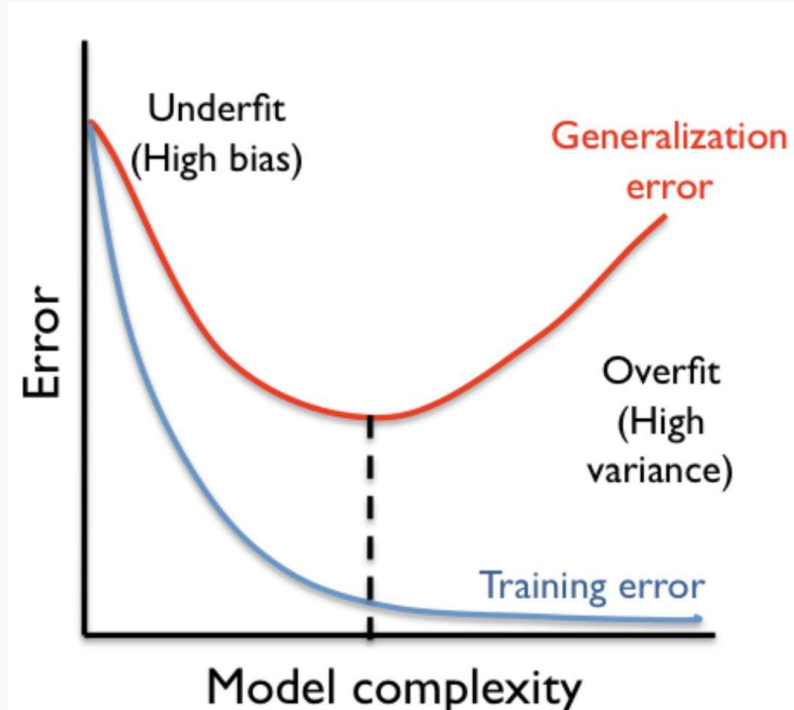
- El error de aproximación puede ser grande si el modelo S no es adaptado
- El error de estimación puede ser grande si el modelo es complejo

Sobre-aprendizaje y sub-aprendizaje



- Según la complejidad del modelo (por ejemplo, tiempo de entrenamiento) se observa un comportamiento diferente
- Los modelos poco complejos son aprendidos fácilmente pero **el error de aproximación** puede ser grande (sub-aprendizaje)
- Los modelos muy complejos pueden tener el objetivo correcto pero un gran **error de estimación** (sobre-aprendizaje)

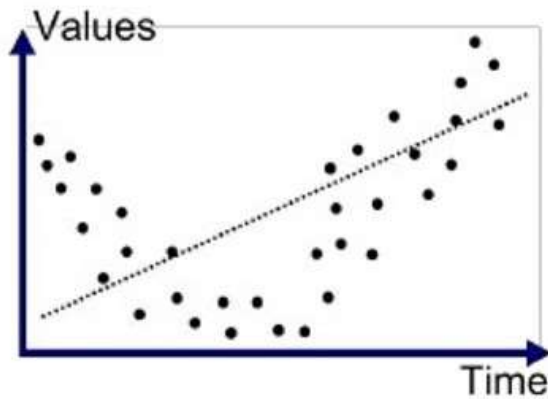
Sobre-aprendizaje: Problema



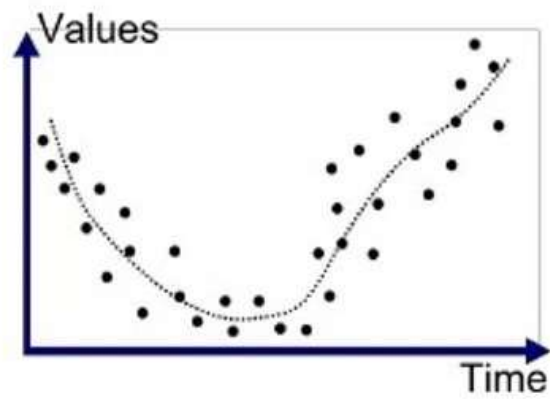
Error y riesgos

- El **riesgo empírico** (error en el conjunto de entrenamiento) disminuye con el aumento de la complejidad del modelo
- El **riesgo real** (error en observaciones de un nuevo conjunto) es muy diferente. ¡Tenemos un **problema de generalización**!
- Sobre-aprendizaje : los parámetros aprendidos son demasiado específicos para el conjunto de entrenamiento
- Se debe usar un criterio diferente al error en el conjunto de entrenamiento

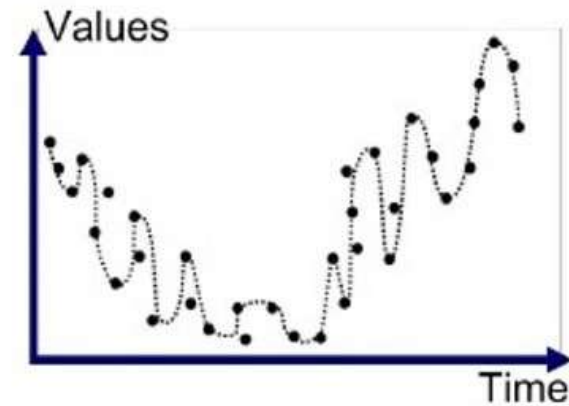
Sobre-aprendizaje: Complejidad



Underfitted



Good Fit/Robust



Overfitted

Complejidad

- Si el modelo es demasiado simple, entonces ya no sigue los datos
- Si el modelo es demasiado complejo, el modelo aprende todas las irregularidades del conjunto de datos D_n
- Ejemplo : si el modelo es el de la curva del medio más una componente de ruido no considerada en las variables, el modelo de la derecha aprende ese ruido

Sobre-aprendizaje: Regularización

Solución para combatir este problema de no generalización: **regularización**

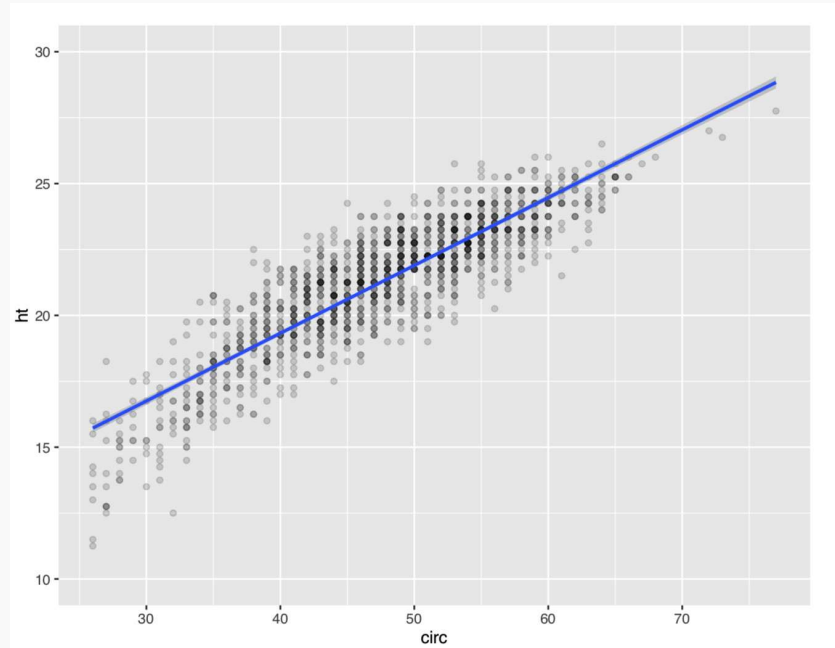
Principio

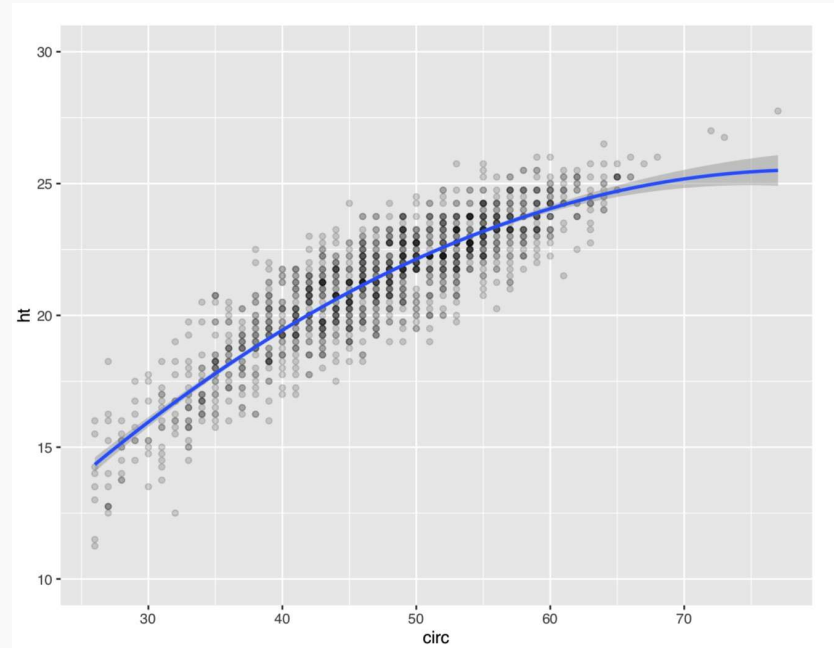
- El riesgo empírico de un estimador seleccionado de una familia de funciones dadas los datos está sesgado
- Al agregar una penalización en relación con la complejidad del modelo f_θ , podemos disminuirla y reducir el sobre-aprendizaje:

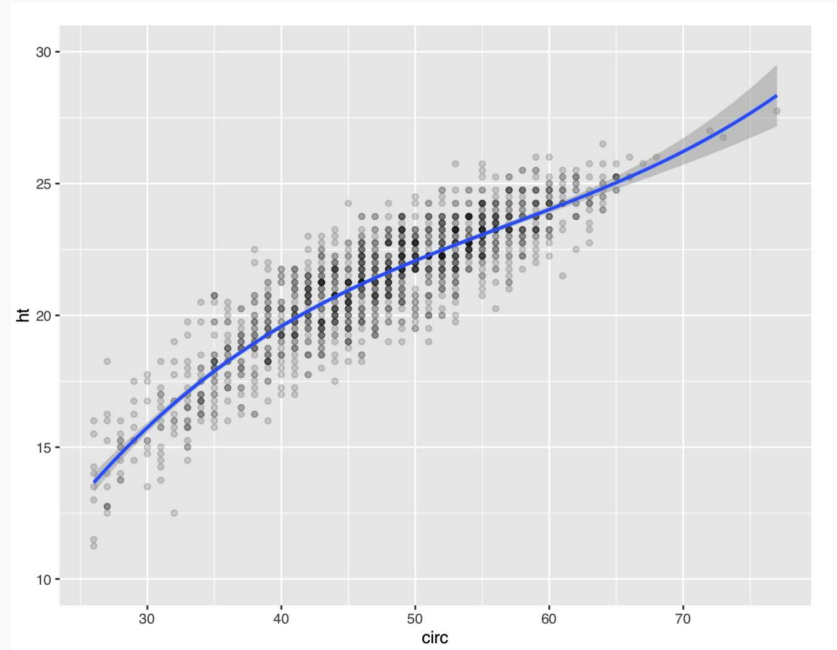
$$\mathcal{R}_n(f_\theta) \rightarrow \mathcal{R}_n(f_\theta) + \text{pen}(\theta)$$

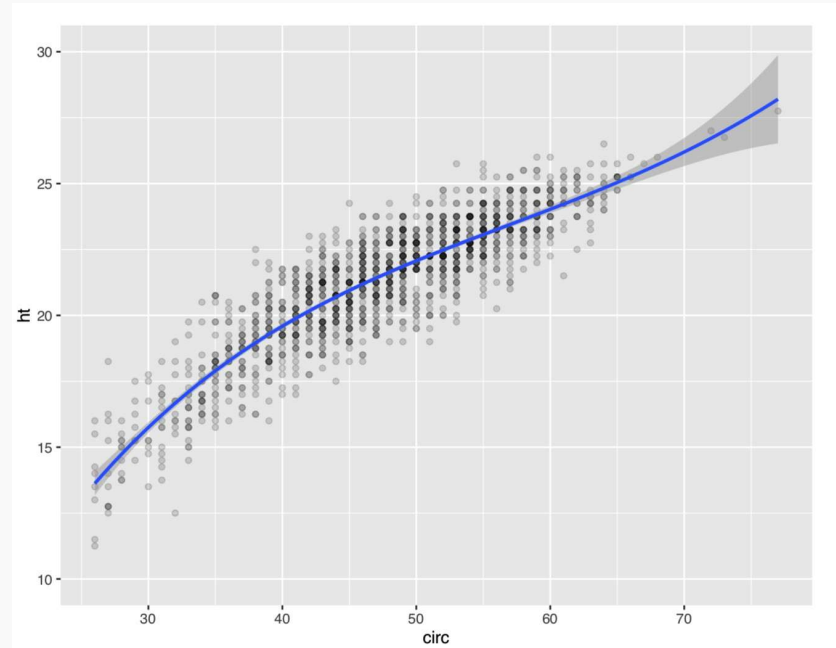
- Entonces, en el riesgo:

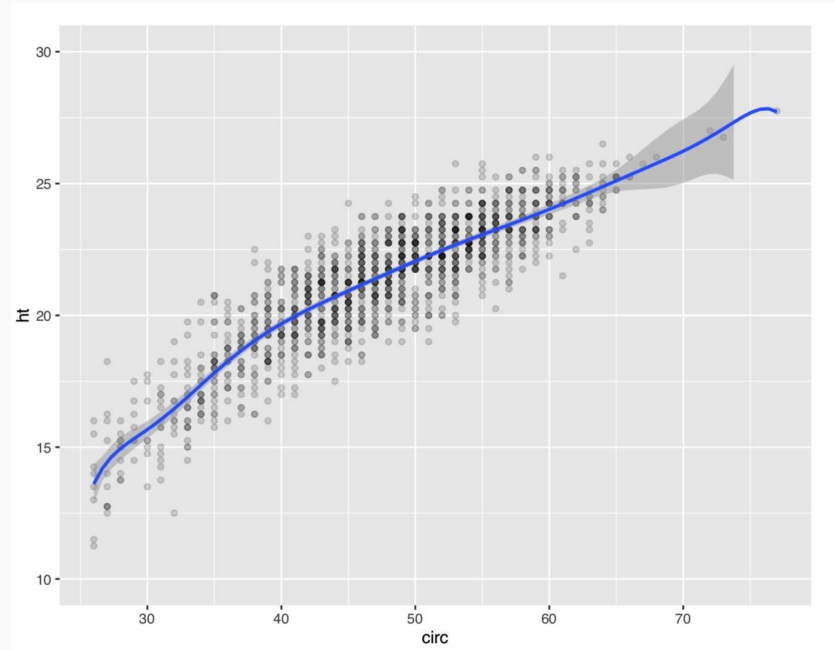
$$\arg \min_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f_\theta(\mathbf{X}_i)) + \text{pen}(\theta)$$



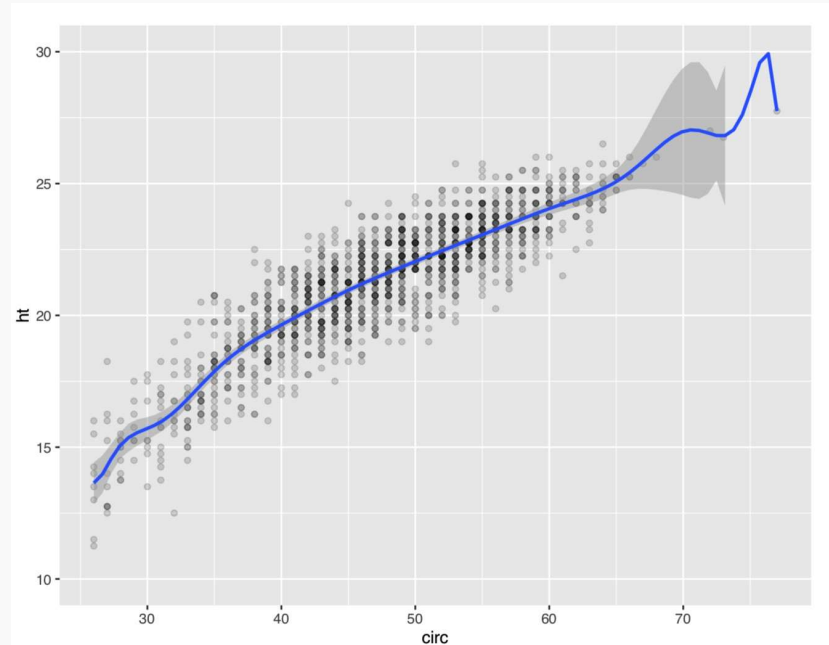








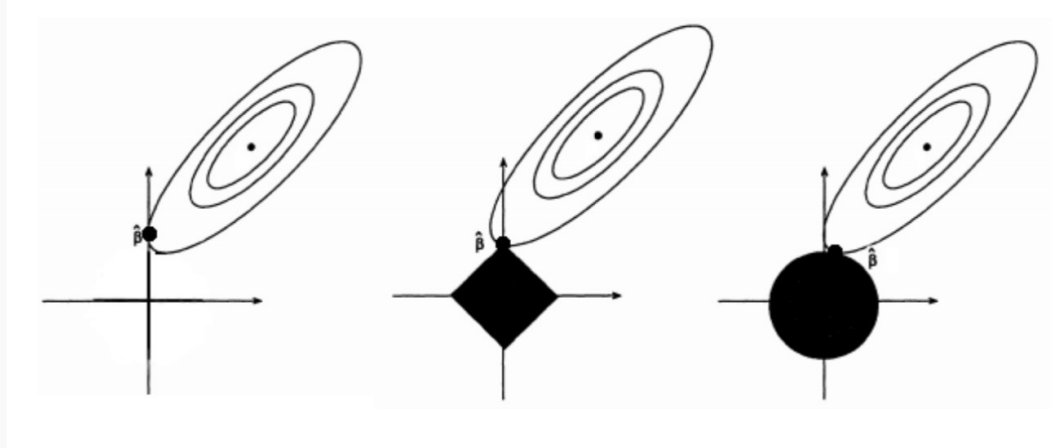
Sobre-aprendizaje: Regularización



Regularización

- Principio de parsimonia (navaja de Occam): cuanto más simple y funcione un modelo, mejor.
- Permite no tener en cuenta tantas particularidades de los datos.
- Intuición: disminuir la norma del modelo o su número de coeficientes, número de ramas del grafo (poda)

Sobre-aprendizaje: Regularización

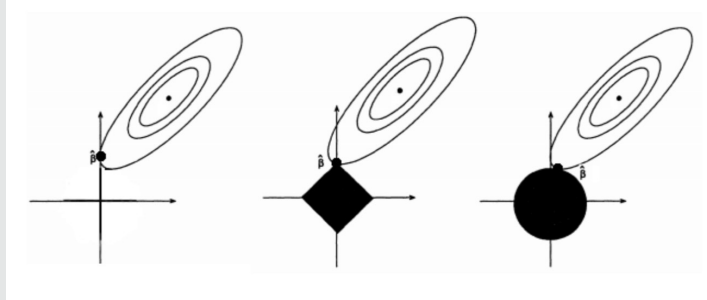


Regularizaciones clásicas

- AIC: $pen(\vartheta) = \lambda ||\vartheta||_0$ (no convexa, parsimoniosa, poco utilizada)
 - Ridge: $pen(\vartheta) = \lambda ||\vartheta||_2$ (convexa, no parsimoniosa)
 - Lasso: $pen(\vartheta) = \lambda ||\vartheta||_1$ (convexa, parsimoniosa)
 - Elastic Net: $pen(\vartheta) = \lambda_1 ||\vartheta||_1 + \lambda_2 ||\vartheta||_2$ (convexa, parsimoniosa)
-
- Optimización simple si el costo (la regularización) es convexo
 - **Necesidad de especificar los λ** , que se convierten en nuevos hiperparámetros

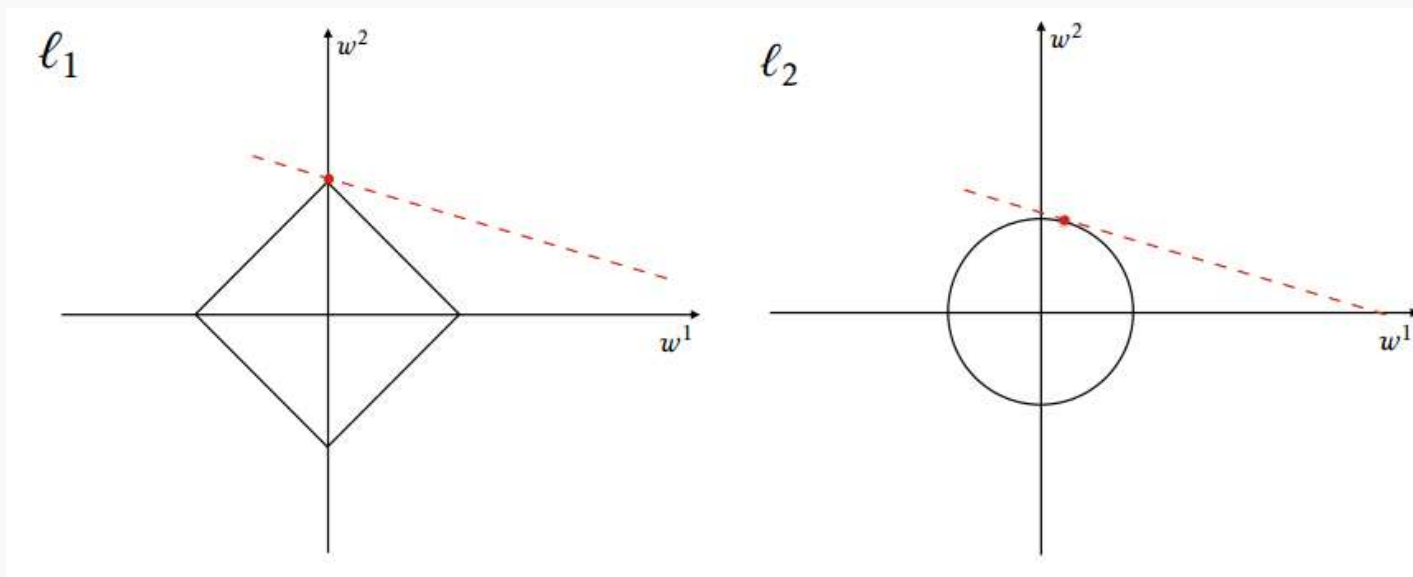
Intuición de la parsimonia

Lasso induce parsimonia

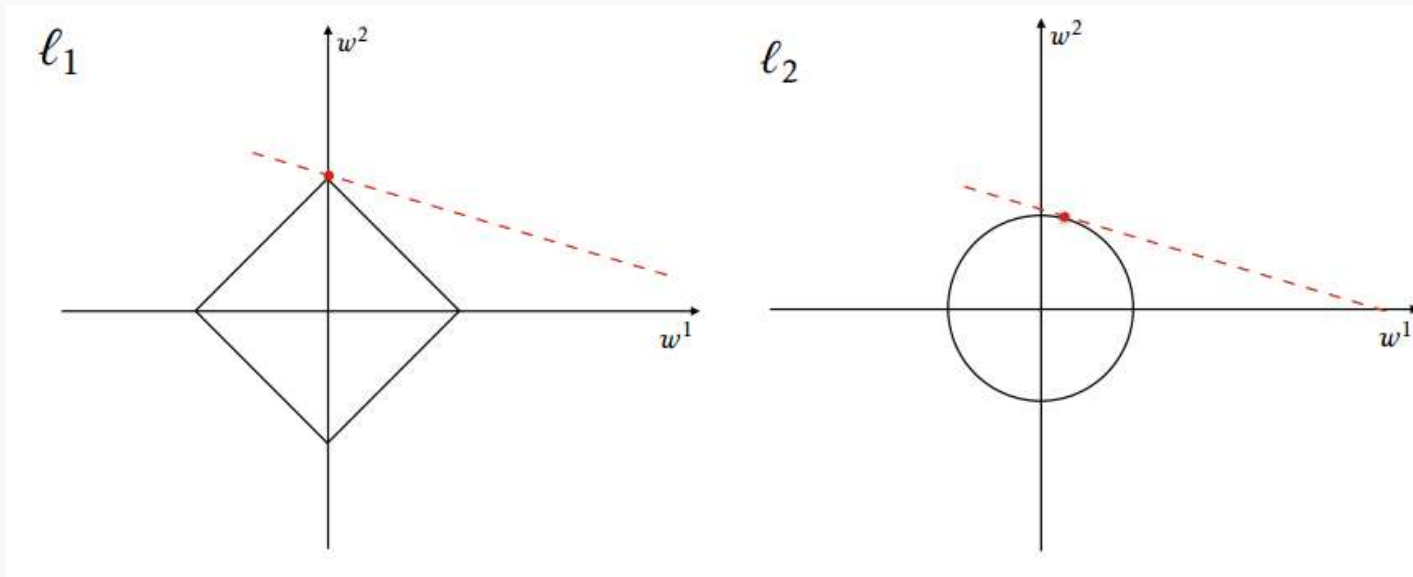


En negro $B^n = \{x \in \mathbb{R}^d \mid \|x\|_n < 1\}$ para $n = 0, 1, 2$ en \mathbb{R}^2

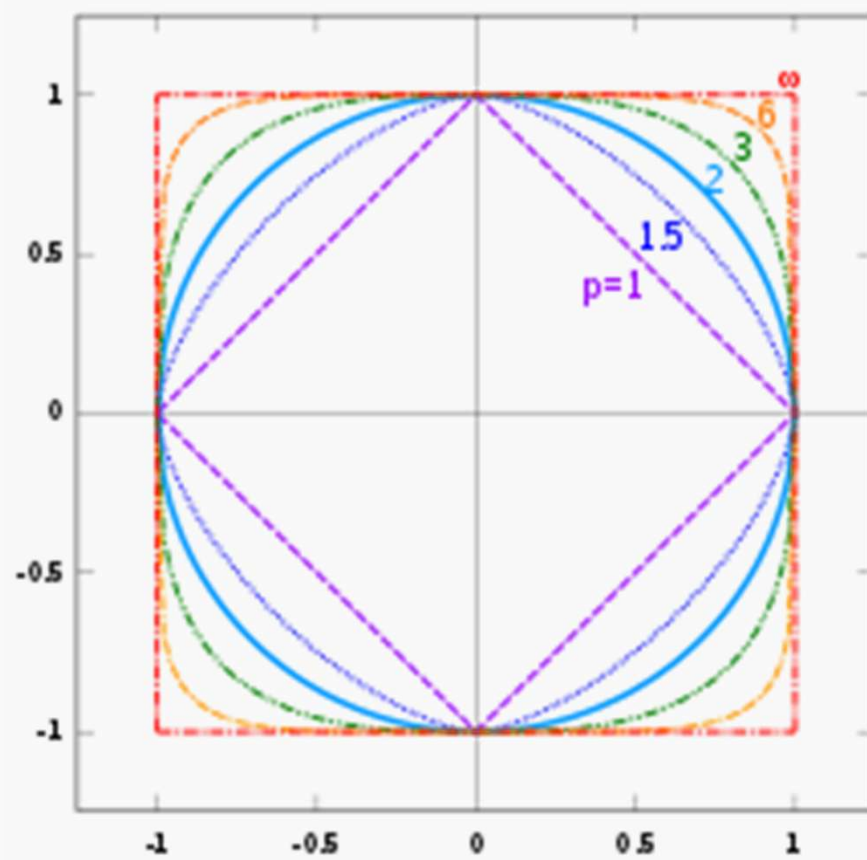
- En dimensiones grandes, la mayoría de B^1 se concentra en los ejes. Esto equivale a tener valores nulos para otros ejes.

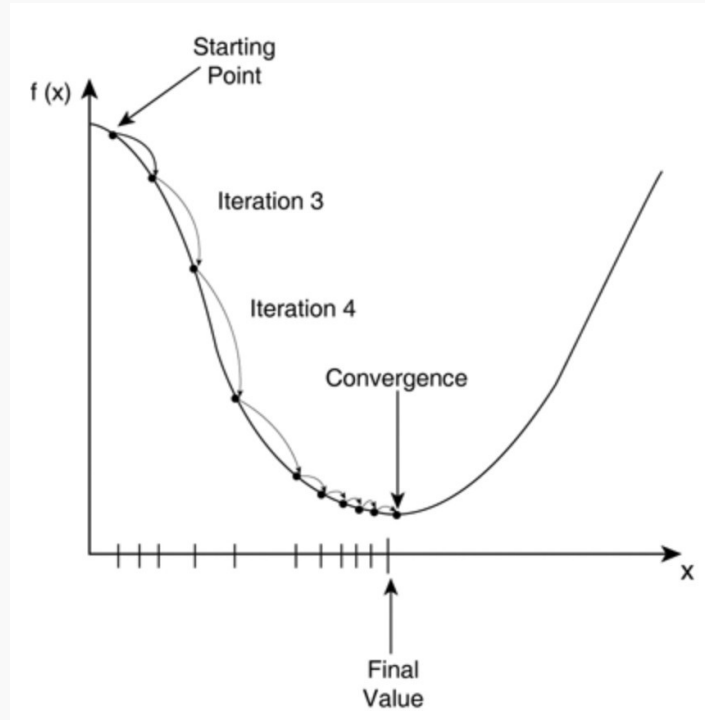


Intuición de la parsimonia : normas



Ejercicio: Hemos visto la bola 1, y la bola 2. ¿Que forma tiene la bola infinita?

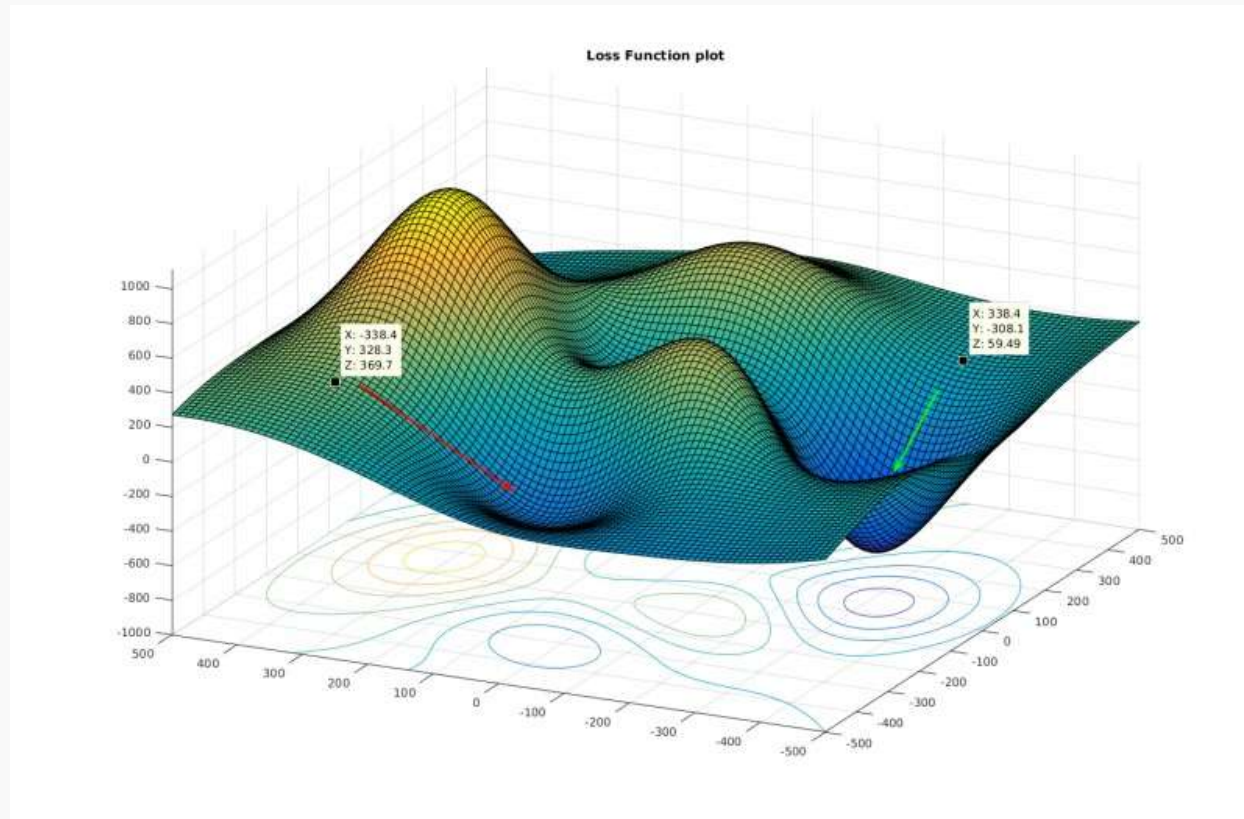




Optimización de la función de costo

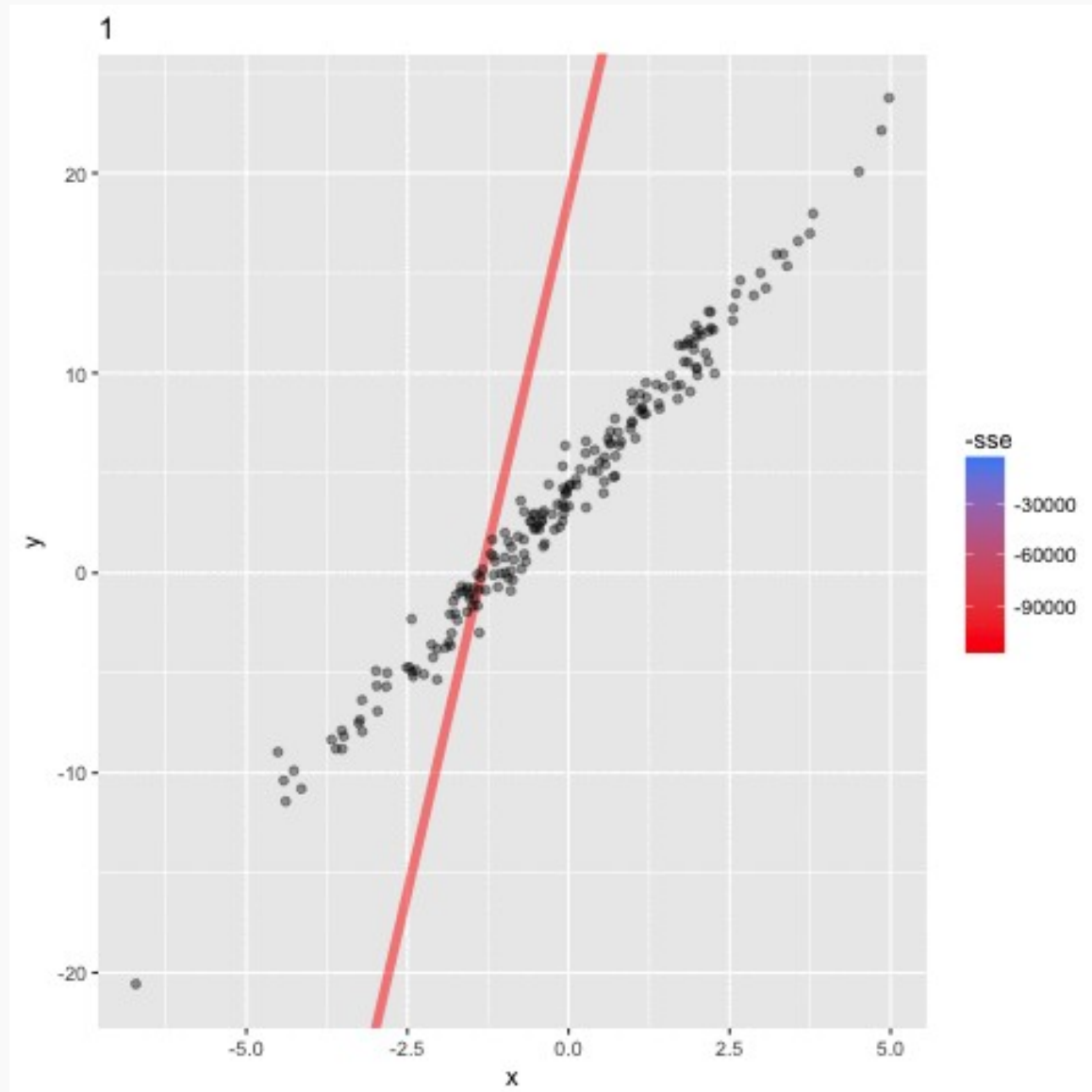
- Sirve para converger al valor mínimo de la función de costo en el conjunto de datos de entrenamiento
- Mejor caso: rápido y preciso
- A menudo se hacen aproximaciones para ser más rápidos

Visualización de la función de costo



Se puede visualizar el valor de la función de costo como una superficie:

- Los valores de los parámetros θ varían en el plano, y el valor de la función $\ell(D_n; \theta)$ varía en altura.
- La convergencia se produce cuando se tienen parámetros que están en un hueco de esta superficie (mínimo local o global, dependiendo del modelo)



Optimización : Descenso del Gradiente Estocástico

Descenso del gradiente

Después de cada cálculo de la función de costo $\ell(Y_i, f_\theta(\mathbf{X}_i); \theta)$, se calcula el gradiente de esta función para actualizar los parámetros θ :

$$\theta \leftarrow \theta - \alpha * \nabla_\theta \ell(Y_i, f_\theta(\mathbf{X}_i); \theta)$$

Ejemplo

- Sean $f_\theta(\mathbf{X}) = \theta^T \mathbf{X} = \sum_k^d \theta_k X^{(k)}$ y $\ell(Y, f_\theta(\mathbf{X})) = \frac{1}{2}(Y - f_\theta(\mathbf{X}))^2$

$$\nabla_\theta \ell(Y_i, f_\theta(\mathbf{X}_i); \theta) = \begin{pmatrix} \frac{\partial}{\partial \theta_1} \\ \vdots \\ \frac{\partial}{\partial \theta_d} \end{pmatrix} \cdot \ell(Y_i, f_\theta(\mathbf{X}_i); \theta) = \begin{pmatrix} X_i^{(1)}(Y_i - f_\theta(\mathbf{X}_i)) \\ \vdots \\ X_i^{(d)}(Y_i - f_\theta(\mathbf{X}_i)) \end{pmatrix}$$

Optimización : Descenso del Gradiente Estocástico

Descenso del gradiente

Después de cada cálculo de la función de costo $\ell(Y_i, f_\theta(\mathbf{X}_i); \theta)$, se calcula el gradiente de esta función para actualizar los parámetros θ :

$$\theta \leftarrow \theta - \alpha * \nabla_\theta \ell(Y_i, f_\theta(\mathbf{X}_i); \theta)$$

Ejemplo

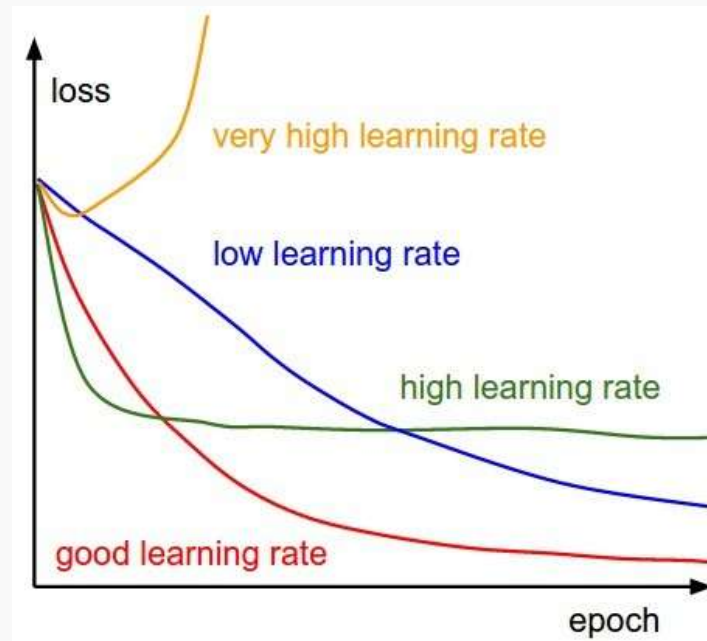
- Sean $f_\theta(\mathbf{X}) = \theta^T \mathbf{X} = \sum_k^d \theta_k X^{(k)}$ y $\ell(Y, f_\theta(\mathbf{X})) = \frac{1}{2}(Y - f_\theta(\mathbf{X}))^2$

$$\begin{pmatrix} \theta_1 \\ \vdots \\ \theta_d \end{pmatrix} \leftarrow \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_d \end{pmatrix} - \alpha * \begin{pmatrix} X_i^{(1)}(Y - f_\theta(\mathbf{X}_i)) \\ \vdots \\ X_i^{(d)}(Y - f_\theta(\mathbf{X}_i)) \end{pmatrix}$$

Optimizaci3n : Importancia de la tasa de aprendizaje

Learning rate

- Un α demasiado peque~no no avanza en el aprendizaje
- Un α demasiado peque~no alarga el tiempo de entrenamiento
- Un α demasiado grande no permite alcanzar el m3nimo (damos vueltas alrededor del agujero de la superficie)
- Un α demasiado grande no permite nada
- Una soluci3n: disminuir α con el tiempo

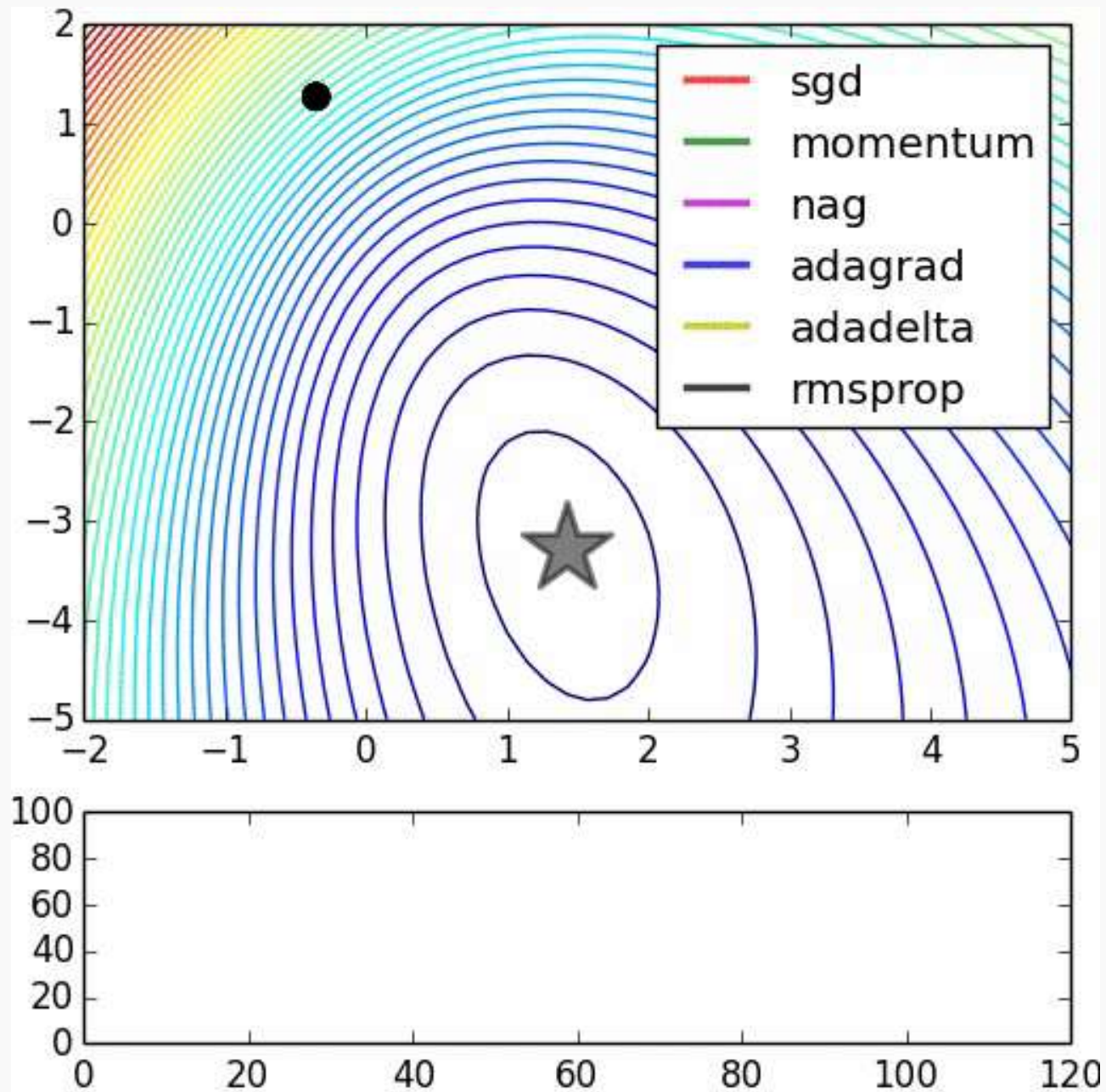


Algoritmos de descenso del gradiente

Existen otros algoritmos existentes que se basan en un descenso del gradiente estocástico (SGD) con especificidades para mejorar su eficacia:

- Descenso del gradiente estocástico con momento
- Gradiente acelerado de Nestorov (NAG)
- Gradiente adaptativo (AdaGrad)
- Adam
- RMSprop

Finalmente, también existen otros métodos más clásicos de descenso: BFGS, L-BFGS, Quasi Newton, ...



Outline : Evaluación de la predicción

Marco global

Generalidades

Contexto teórico

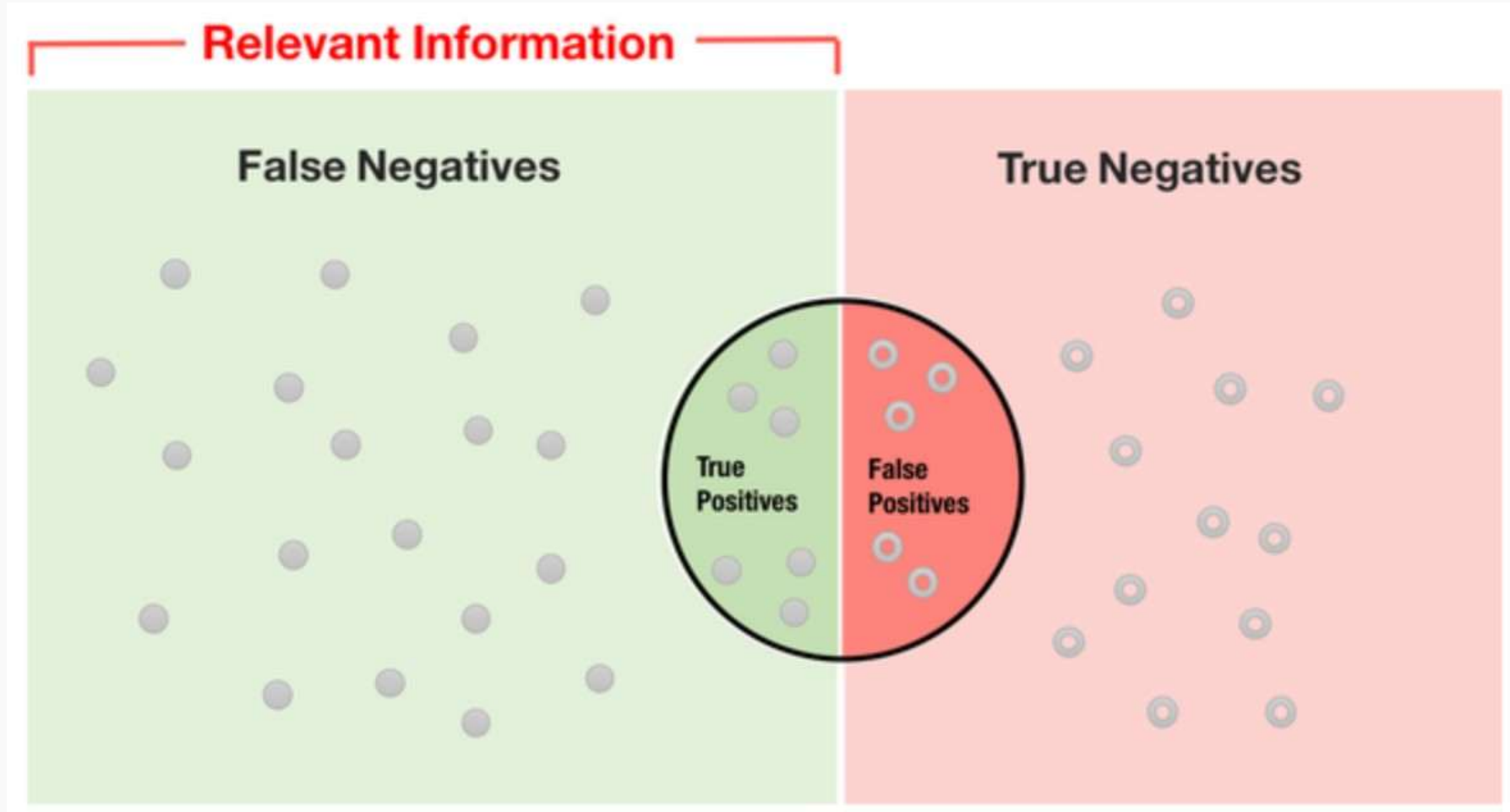
Aprendizaje

Evaluación de la
predicción

Métricas de clasificación
Métricas de regresión

Muestro para validación

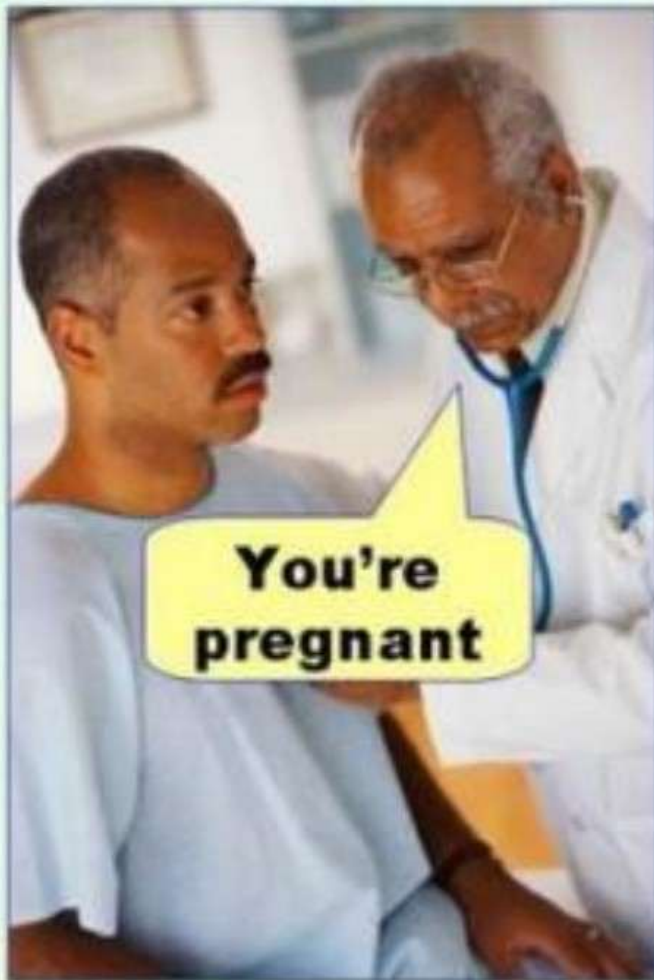
Evaluación de la predicción: Verdaderos y falsos positivos



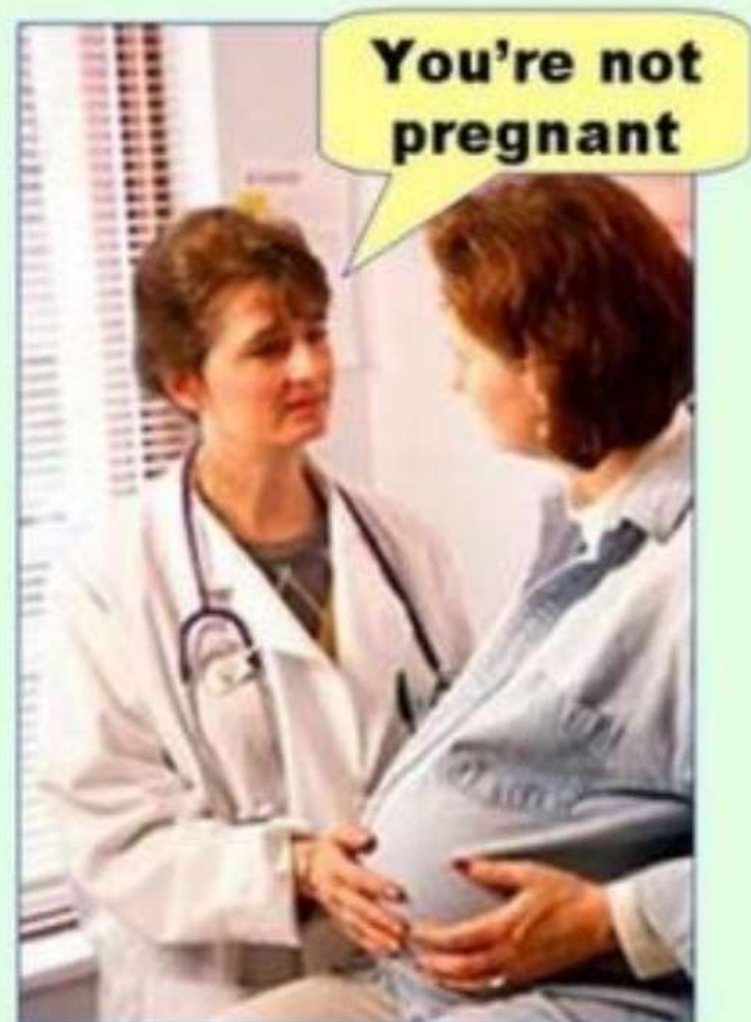
Preste **atención**

- Si el conjunto de datos no está equilibrado
- La importancia que se le da a los diferentes tipos de errores

Type I error
(false positive)



Type II error
(false negative)



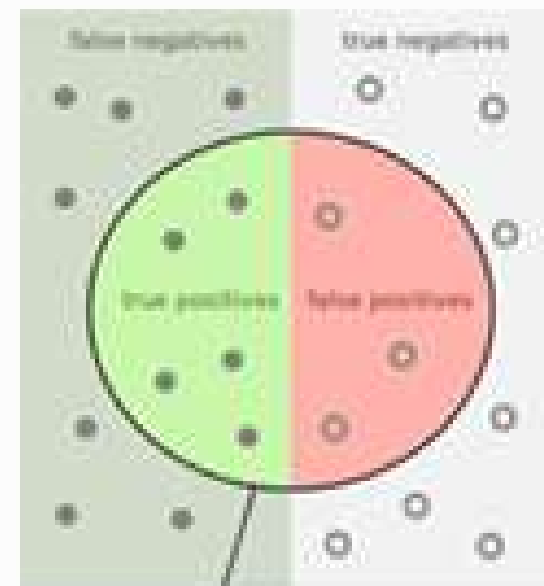
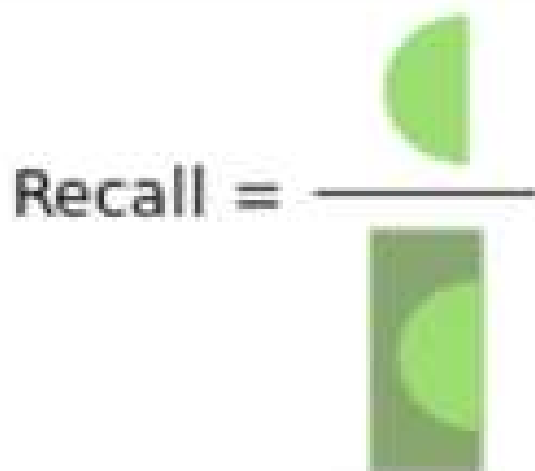
		Model prediction / Test result	
		Positive / Predicted positive	Negative / Predicted Negative
Actual / Ground truth / target / condition / Label	Positive (P)	True Positive (TP), hit	False Negative (FN), type II error, miss, Underestimation
	Negative (N)	False Positive (FP), type I error, false alarm, Overestimation	True negative (TN), correct rejection

Métricas: Recall

Recall: ¿Qué proporción de los positivos reales se clasificaron como positivos?

$$\text{Recall} = \frac{TP}{TP + FN}$$

- ¿Se han clasificado a todos?
- ¿En qué caso esta métrica podría ser la más importante?

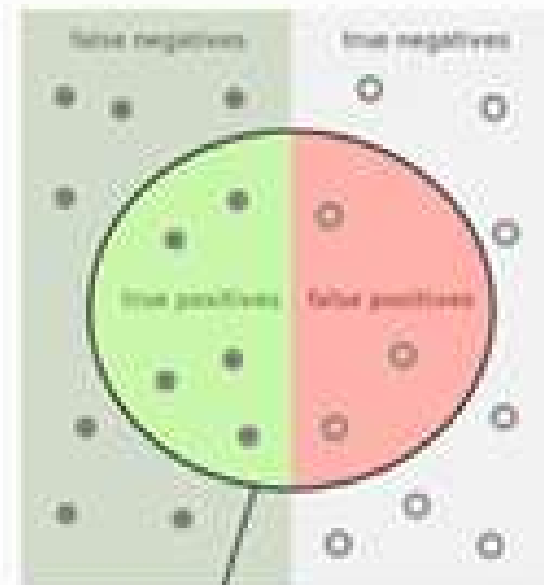
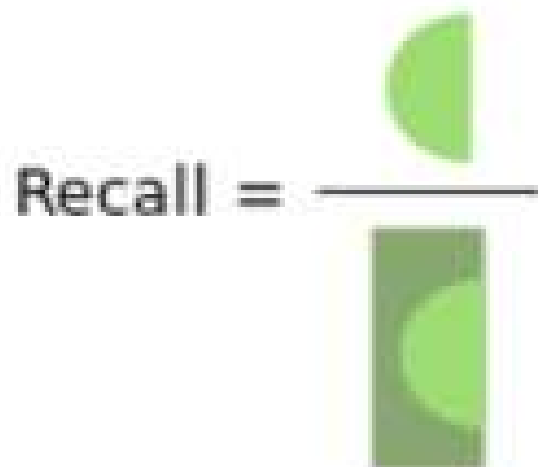


Métricas : Recall

Recall: ¿Qué proporción de los positivos reales se clasificaron como positivos?

$$\text{Recall} = \frac{TP}{TP + FN}$$

- ¿Se han clasificado a todos?
- ¿En qué caso esta métrica podría ser la más importante?
- **Detección masiva de una enfermedad contagiosa**: no queremos perder a ningún contaminado

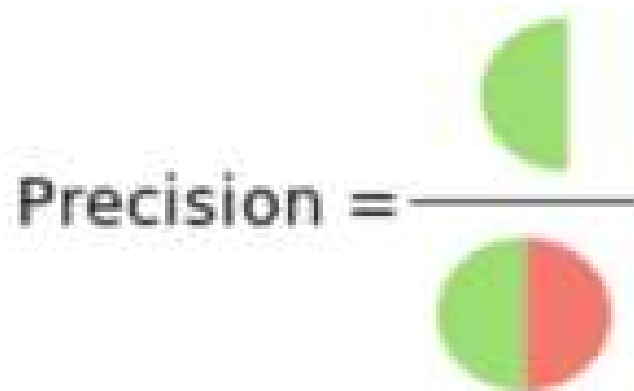


Métricas : Precisión

Precisión: ¿Qué proporción de las predicciones positivas son correctas?

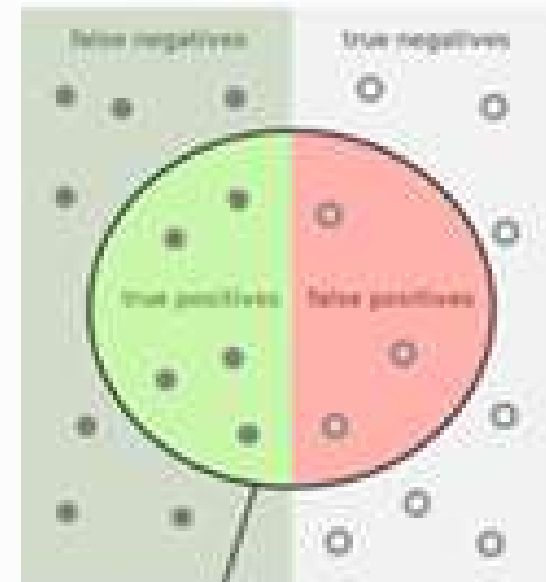
$$\text{Precisión} = \frac{TP}{TP + FP}$$

- ¿Se han clasificado a todos?
- ¿En qué caso esta métrica podría ser la más importante?



The diagram shows a green semi-circle above a horizontal line, and a circle below the line, divided vertically into a green left half and a red right half. This represents the formula: Precision = TP / (TP + FP).

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

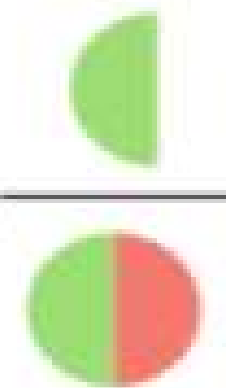


Métricas : Precisión

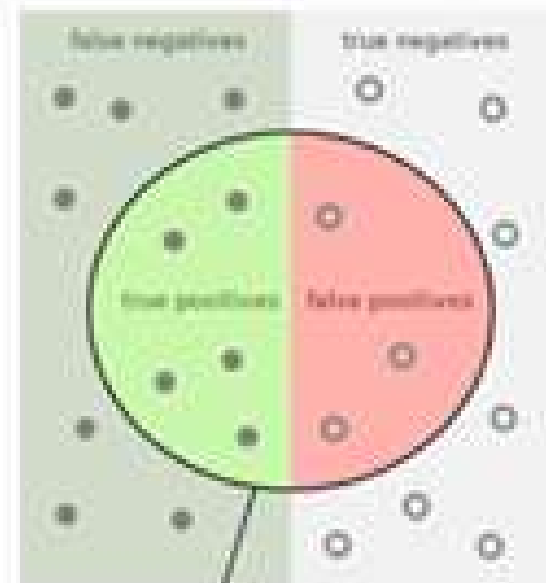
Precisión: ¿Qué proporción de las predicciones positivas son correctas?

$$\text{Precisión} = \frac{TP}{TP + FP}$$

- ¿Se han clasificado a todos?
- ¿En qué caso esta métrica podría ser la más importante?
- **Detección fina de una enfermedad mortal**: no queremos dar ningún tratamiento pesado sin necesidad



Precision = $\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$

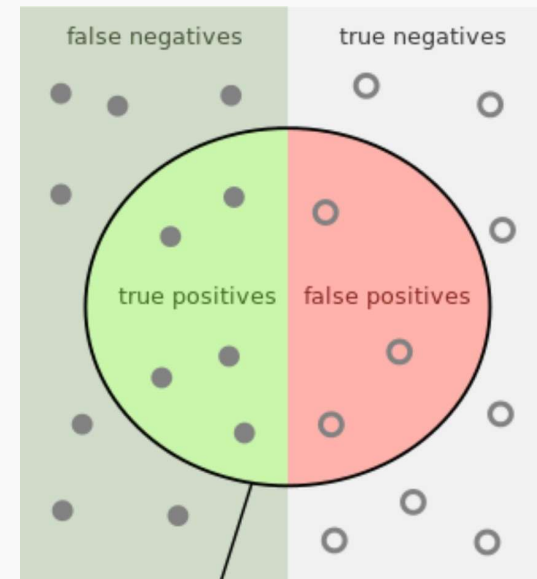
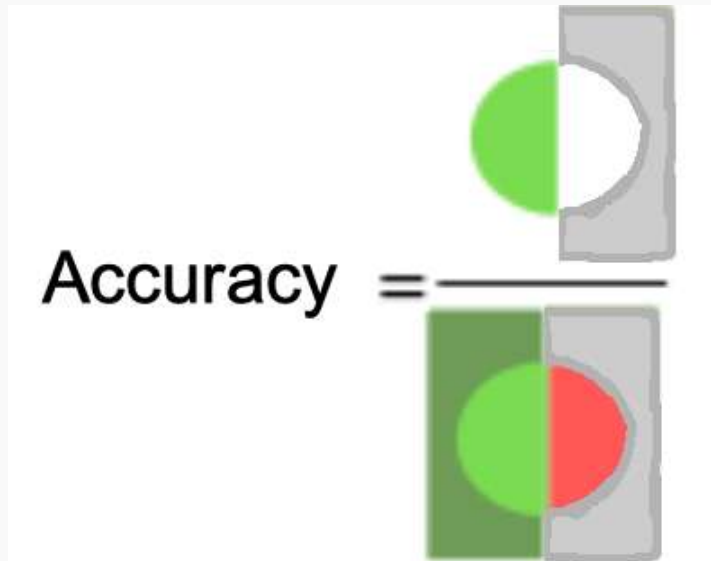


Métricas: Accuracy

Precisión: ¿Qué proporción de las predicciones son correctas?

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

- ¿Qué tan bien funciona en el conjunto de datos global?
- ¿En qué caso esta métrica es completamente inútil?

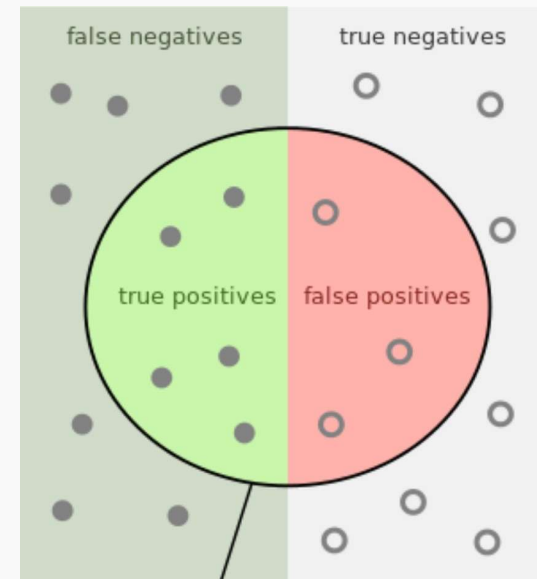
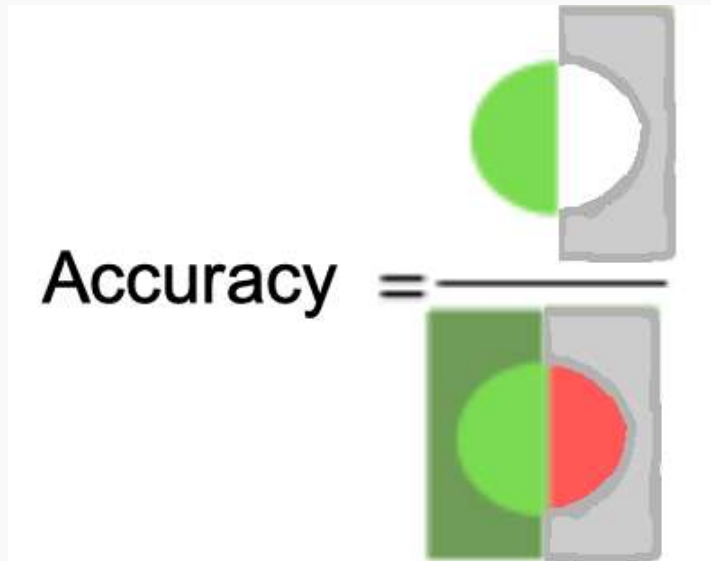


Métricas : Accuracy

Precisión: ¿Qué proporción de las predicciones son correctas?

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

- ¿Qué tan bien funciona en el conjunto de datos global?
- ¿En qué caso esta métrica es completamente inútil?
- **Detección de anomalías:** con 1% de casos positivos, un sistema que predice negativo todo el tiempo da un 99% de precisión...



Métricas : Ejercicio I

Considere 286 mujeres: 201 no tienen reincidencia de cáncer después de 5 años y 85 sí tienen.

Compare los modelos:

- M1: "todas reinciden"
- M2: "ninguna reincide"

TODO

Hacer matrices de confusión, calcular accuracy, precision, recall y F1.

Considere 286 mujeres: 201 no tienen reincidencia de cáncer después de 5 años y 85 sí tienen. Compare los modelos:

M1: Todas reinciden

M1	+	-
+	85	0
-	201	0

Accuracy: $85/286 = 0.3$

Precision: $85/85 = 1$

Recall: 1

F1: $2 \cdot 0.3 / (0.3 + 1) = 0.46$

M2: Ninguna reincide

M2	+	-
+	0	85
-	0	201

Accuracy: $201/286 = 0.7$

Precision: $0/0 = \text{undef}$

Recall: $0/85 = 0$

F1: undef

Métricas : Cost Matrix

La idea: Hay errores mas graves que otras

Poner un peso sobre el tipo de error

A veces yo se cuales errores son más costosos y cuales aciertos son más valiosos.

Clase real	Clase predicha		
	$C(i j)$	clase = +	clase = -
	clase = +	$C(+ +)$	$C(- +)$
	clase = -	$C(+ -)$	$C(- -)$

$C(i|j)$: Costo de clasificar un objeto como clase j dado que es clase i

Métricas : Cost Matrix

La idea: Hay errores mas graves que otras

Poner un peso sobre el tipo de error

A mayor costo
peor el modelo.

Matrix Costo	Clase predicha		
	C(i j)	+	-
Clase real	+	-1	100
	-	1	0

Modelo M1	Clase predicha		
Clase real		+	-
	+	150	40
	-	60	250

Accuracy(M1) = 0.8

$$C(M1) = -1*150+100*40+1*60+0*250 = 3910$$

Modelo M2	Clase predicha		
Clase real		+	-
	+	250	45
	-	5	200

Accuracy(M2) = 0.9

$$C(M2) = -1*250+100*45+1*5+0*200 = 4255$$

Diferentes métricas según lo que se busca

-

$$\text{Recall} = \frac{TP}{TP + FN}$$

-

$$\text{Precision} = \frac{TP}{TP + FP}$$

-

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

-

$$F1 = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2TP}{2TP + FP + FN}$$

Más generalmente: $\frac{1}{F_\beta} = \frac{1}{1+\beta^2} \left(\beta^2 \frac{1}{\text{Recall}} + \frac{1}{\text{Precision}} \right)$

- Curva ROC, AUC,...
- top-k accuracy, balanced accuracy

More infos: https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics

Si hay mas de una clase, hay que agregar las métricas per clase:

- **Micro-averaging:** computar métrica para cada clase y luego promediar
- **Macro-averaging:** crear matriz de confusión binaria para cada clase, combinar las matrices y luego evaluar

		<i>gold labels</i>			
		urgent	normal	spam	
<i>system output</i>	urgent	8	10	1	precision_u = $\frac{8}{8+10+1}$
	normal	5	60	50	precision_n = $\frac{60}{5+60+50}$
	spam	3	30	200	precision_s = $\frac{200}{3+30+200}$
		recall_u = $\frac{8}{8+5+3}$	recall_n = $\frac{60}{10+60+30}$	recall_s = $\frac{200}{1+50+200}$	

Figure 4.5 Confusion matrix for a three-class categorization task, showing for each pair of classes (c_1, c_2) , how many documents from c_1 were (in)correctly assigned to c_2

Agregación de Métricas II

Class 1: Urgent			Class 2: Normal			Class 3: Spam			Pooled		
	true urgent	true not		true normal	true not		true spam	true not		true yes	true no
system urgent	8	11	system normal	60	55	system spam	200	33	system yes	268	99
system not	8	340	system not	40	212	system not	51	83	system no	99	635

precision = $\frac{8}{8+11} = .42$

precision = $\frac{60}{60+55} = .52$

precision = $\frac{200}{200+33} = .86$

microaverage
precision = $\frac{268}{268+99} = .73$

macroaverage
precision = $\frac{.42+.52+.86}{3} = .60$

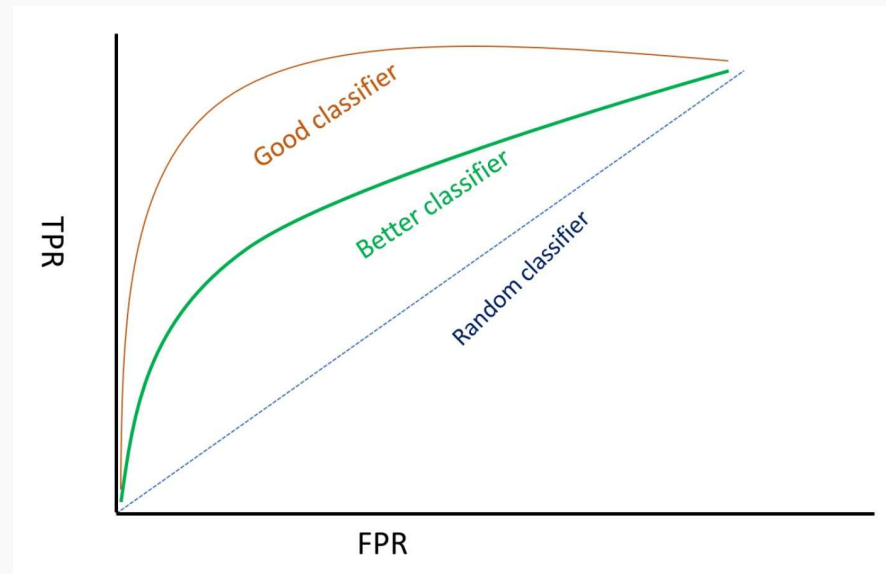
Figure 4.6 Separate contingency tables for the 3 classes from the previous figure, showing the pooled contingency table and the microaveraged and macroaveraged precision.

- Los micro-promedios son dominados por las clases más frecuentes.
- Los macro-promedios pueden sobre-representar a clases minoritarias.

Receiver operating characteristic (ROC)

Definicion

Ilustra el rendimiento de un sistema clasificador binario a medida que varía su **umbral de discriminación**. Se crea trazando la fracción de verdaderos positivos de los positivos (TPR = true positive rate) frente a la fracción de falsos positivos de los negativos (FPR = false positive rate), en varios ajustes de umbral.



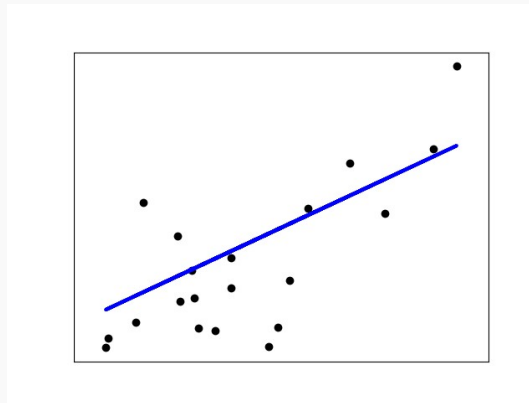
El AUC (area under curve) se utiliza porque mas grande significa mejor.

Evaluación de la predicción: distancia

Diferentes funciones para calcular el rendimiento de una regresión

- Error medio absoluto: $\frac{1}{n} \sum_{i=1}^n |Y_i - f(\mathbf{X}_i)|$
- Error cuadrático medio: $\frac{1}{n} \sum_{i=1}^n |Y_i - f(\mathbf{X}_i)|^2$
- Error medio mediano: $\text{mediana}(|Y_1 - f(\mathbf{X}_1)|, \dots, |Y_n - f(\mathbf{X}_n)|)$
- Coeficiente de determinación R^2 :

$$1 - \frac{\sum_{i=1}^n |Y_i - f(\mathbf{X}_i)|^2}{\sum_{i=1}^n |Y_i - \bar{Y}|^2}$$



R^2 representa la **proporción de la varianza explicada** por el modelo

Validación Cruzada (Cross-Validation)

Principio

- Separación del conjunto de datos \mathcal{D} en V conjuntos \mathcal{D}_v de tamaños similares
- Para cada $v \in \llbracket 1; V \rrbracket$:
 - Tomar $\mathcal{D}^{-v} = \mathcal{D} \setminus \mathcal{D}_v$
 - Entrenar \hat{f}^{-v} en \mathcal{D}^{-v}
 - Calcular $\mathcal{R}^{-v}(\hat{f}^{-v}) = \frac{1}{n_v} \sum_{(\mathbf{X}_i, Y_i) \in \mathcal{D}_v} l(Y_i, \hat{f}^{-v}(\mathbf{X}_i))$
- Calcular el **riesgo general** : $\mathcal{R}^{CV}(\hat{f}) = \frac{1}{V} \sum_{i=1}^V \mathcal{R}^{-v}(\hat{f}^{-v})$

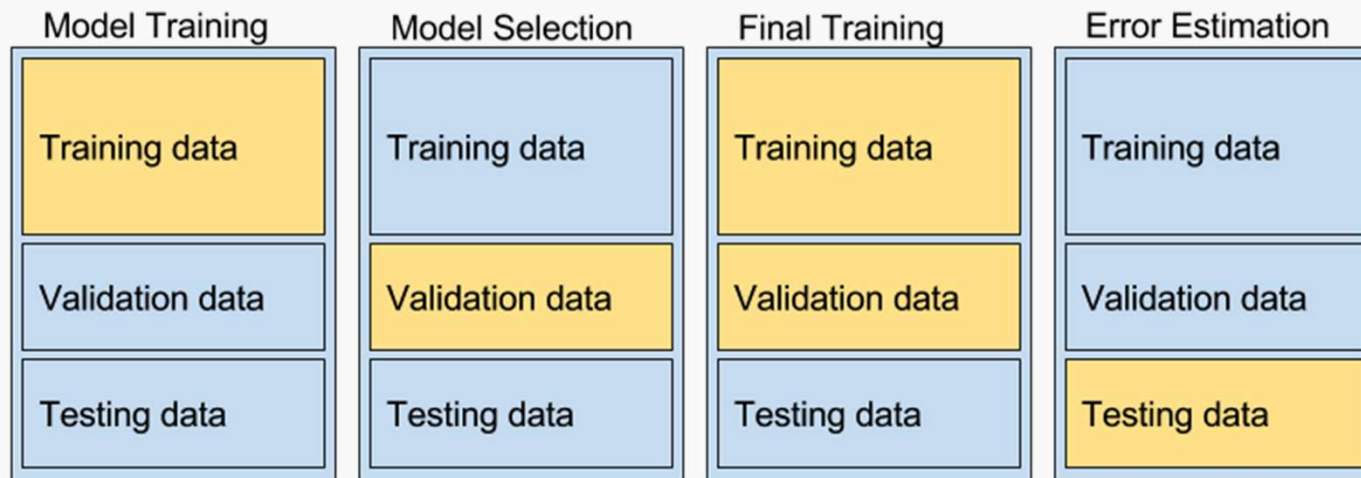
Train	Train	Test
Test	Train	Train
Train	Test	Train

Conjuntos de validación y prueba (Holdout)

- La validación cruzada requiere V entrenamientos. ¿Existe un método menos costoso?

Separación del conjunto de datos D en 3 conjuntos

- El **conjunto de entrenamiento** se utiliza para entrenar el clasificador
- El **conjunto de validación** proporciona una idea del poder de generalización del modelo entrenado. Puede usarse para detener el entrenamiento
- El **conjunto de prueba** es un conjunto independiente utilizado para probar el rendimiento del clasificador, no interactúa con el entrenamiento del clasificador



70/10/20 o 70/15/15 son buenas proporciones para train/val/test

Tamaño de la partición

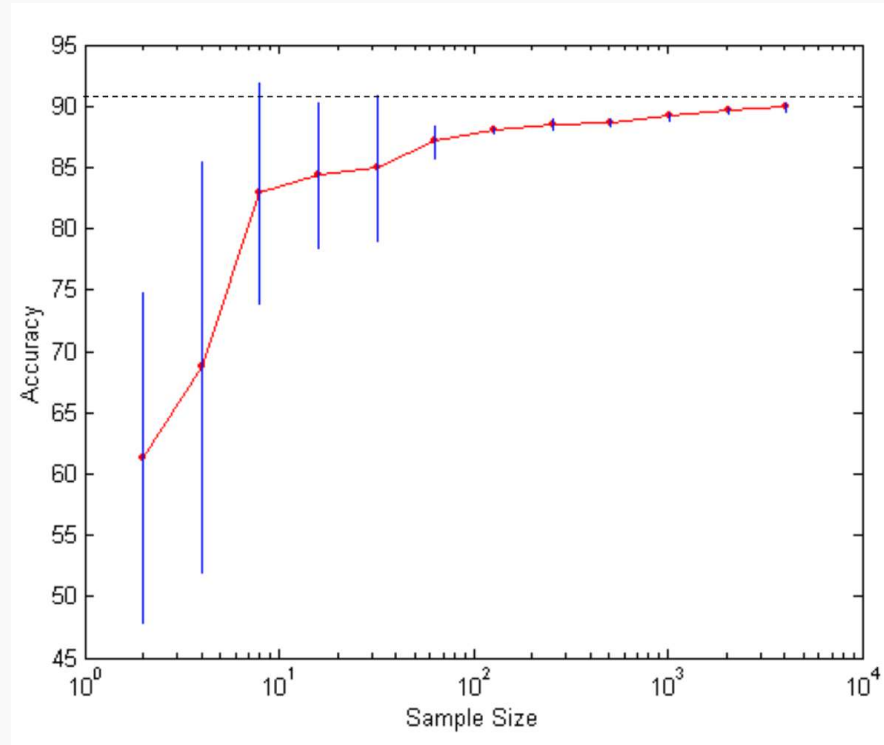


Figure 1: Performancias de un modelo segun la cantidad de datos bista

- La evaluaci3n puede variar mucho segun las particiones escogidas.
- Training muy pequen3o \Rightarrow modelo sesgado.
- Testing muy pequen3o \Rightarrow accuracy poco confiable.

Questions?

References i